Sun QFS and Sun Storage Archive Manager 5.1

ORACLE®

Home

Sun QFS and Sun Storage Archive Manager (SAM-QFS) 5.1 Information

This PDF file contains documentation for the Sun QFS file system and Sun Storage Archive Manager (SAM-QFS) 5.1 release.

Because of the migration of SAM-QFS documentation from wikis.sun.com to OTN, some links in this PDF file might not work. You will also need to access man pages from the product software rather than through this PDF file.

Printable Pages

The following pages are similar to what previous product releases provided as "books." The following is a list of available books and the corresponding page numbers in the PDF file.

Getting Started

- About QFS and SAM-QFS
  - More About QFS
  - More About SAM-QFS
- Sun QFS and SAM-QFS 5.1 Release Notes
- New Features in SAM-QFS 5.1
- Planning Your Environment

Installing

- Quick Start
- Preparing for Installation
- Release Package Contents
- Installing SAM-QFS (Archiving and File System)
- Installing QFS (File System Only)
- Installing SAM-QFS Manager
- Performing Additional SAM-QFS Configuration
- Upgrading Sun QFS and SAM-QFS

Removing Software

- Uninstalling SAM-QFS Manager

Configuring File Archiving (SAM-QFS)

Getting the Storage Devices Ready

- Configuring Storage Devices for Archiving
- Creating Parameters Files for Network Attached Automated Libraries
- Checking the Drive Order in Libraries
- Populating the Catalog

Managing the Storage Hardware and Media

- Managing Automated Libraries and Manually Loaded Drives
- Managing Vendor-Specific Libraries

Managing the Archiving Components

- About Archiving
    - Configuring the Archiver
    - Archive Directives (archiver.cmd)
    - Archive Set Directives (archiver.cmd)
- About Releasing
    - Configuring the Releaser
- About Staging
    - Configuring the Stager
- About Recycling
    - Configuring the Recycler
- Advanced SAM-QFS Topics
- Using the Sun SAM-Remote Software

Configuring File Systems (QFS)

Understanding QFS File Systems

- File System Overview
- File System Design Basics
- File Allocation Methods

Configuring QFS File Systems

- About the Master Configuration File
- Configuring the File System
- Configuring a Shared File System
- Resizing File Systems

Managing QFS File Systems

- Performing Operations
- Administering File System Quotas
- Advanced File System Topics

Using

- Using SAM-QFS on Linux Clients
- Using SAM-QFS With Solaris Cluster
- Using the samu Operator Utility
- Using QFS File Systems With SANergy (SAN-QFS)

Support

- Disaster Planning and Recovery
- SAM-QFS Troubleshooting
- Best Practices

Reference Information

- Mount Options in a Shared File System
- SAM-QFS Glossary

Sun QFS and SAM-QFS 5.1 Release Notes

Contents

Sun QFS and SAM-QFS 5.1 Release Notes

What's New in the SAM-QFS 5.1 Release?

The SAM-QFS 5.1 release provides the foundation for the next generation of file system and storage archive management products. This release improves stability and performance over the previous SAM-QFS products. Although no significant new features are provided, several key enhancements are intended to ensure the safety of your data.

Specific fundamental improvements include the following:

- I/O path – Several bugs were fixed that relate to memory mapping I/O.
- Archiver operations – Several bugs were fixed that improve archiver performance and operations.
- Recycler operations – Several bugs were fixed that improve recycler performance and operations.
- Stager operations – Several bugs were fixed that improve stager performance and operations.
- Sideband database consistency – Several bugs were fixed that improve the consistency of the sideband database.

- Rolling upgrades – If you are already running SAM-QFS 5.0, you can upgrade individual shared file system clients to SAM-QFS 5.1 without taking down the rest of the file system. For more information, see Support for Rolling Upgrades in a Shared Environment.
- Improved remote disk archive performance, especially in a high-latency WAN environment – The remote disk archive write size on the client is now set to the value of the `blksize` parameter set in the `/etc/opt/SUNWsamfs/rft.cmd` file on the server. If no value is set, the default is 1 MB.
- Stager direct I/O behavior – The previous default for `dio_min_size` threshold on the stager was set at 32 MB. The threshold was changed to 8 MB to improve performance.

To view an abridged list by product feature of bugs fixed in this release, see Significant Bugs Fixed in This Release.

Also, a few key enhancements were made to the documentation for this release:

- Man pages are now viewable from the documentation wiki. See SAM-QFS Man Pages by Section.
- Information about using SAM-QFS with Solaris Cluster has been significantly enhanced. See Using SAM-QFS With Solaris Cluster.

Operating Systems

Sun QFS and SAM-QFS 5.0 software require the following minimum operating system release:

- Solaris 10, Update 6

In addition, you can use any of the following operating systems as a client in a shared file system:

- Solaris 10, Update 6 for x86 (32-bit)
- Red Hat Enterprise Linux 4.0 (UD-4) for x64 platforms
- Red Hat Enterprise Linux 4.5 for x64 platforms
- SuSE Linux Enterprise Server 9 (service pack 2) for x64 platforms
- SuSE Linux Enterprise Server 10 for x64 platforms
- SuSE Linux Enterprise Server 10 (service pack 2) for x64 platforms

Support for Rolling Upgrades in a Shared Environment

You can upgrade individual shared file system clients without taking down the rest of the file system. To perform a rolling upgrade, your environment must include one primary metadata server and at least one potential metadata server.

Follow these steps:

1. Upgrade the potential metadata server.
2. Failover to the potential metadata server.
3. Upgrade the primary metadata server.
4. Failover to the primary metadata server.
5. Upgrade the clients.

> Note -
> At any given time, the metadata server and the clients can only be one release apart.

Significant Bugs Fixed in This Release

File System Manager

Bug Number	Description
6759875	Directories continue to show in the recovery points list even after recovery points are removed from that directory.
6845375	SUNWfsmgr* packages are not zone-aware.
6903985	When an archive set is used without directives, the GUI fails to display the Policy details.
6838156	Loading log and trace file information causes an unrecoverable error when browsing in the Fault Summary page.
6908407	File System Manager silently fails when indexing recovery points, which leads you to believe valid recovery points exist when they do not.

6916183	File System Manager does not notify you when you try to administer an older, unsupported version of the product.

File Systems

Bug Number	Description
6815281	Using NFS and Hitachi TrueCopy with standalone QFS file system might result in file corruption.
6824029	Attempting a restore of a dump made with `samfsdump -uP` might cause damaged inodes if there are dumped files with partial release set and the file size is less than the partial release size.
6814247	SAM-QFS panics and shows "recursive rw_enter" messages in the `/var/adm/messages` file.
6834508	ACLs not updated correctly on shared clients.
6610921	Forced unmount panics when archiver is running.
6906396	Panic can occur when trying to remove a directory during a shrink.
6881109	When a process on the file system client exceeds quotas when writing a file, if that file is removed, the corresponding data blocks are not reclaimed back to the free space and quota information is not updated.
6862290	`samfsrestore` of a WORM (emul_lite) file system does not retain `retention-period` information.
6855570	Following a `samfsrestore` of a SAM-QFS dump (all metadata, no data) to a new file system shows missing files. Running a `samfsck` on the new file system show exactly the same number of orphan inodes. Restoring a single missing file from the dump to the new file system seems to be successful, however the file later disappears again.
6826981	`schproj` ignores path access permissions.
6869469	When Oracle application binaries reside on shared QFS file systems, management of leases can lead to periodic bursts of CPU activity, which deprive the system of the critical resources.
6900403	Linux clients for sites that have hundreds of clients send send BLOCK_vfsstat_v2 every 5 seconds.
6838989	`samfsconfig` needs to support SVM mirroring.
6836907	System panics while doing `sls -D` on an inode that spans multiple volumes.

Archiver

Bug Number	Description
6282366	If a remote host goes down, the archive database notifies you that there is an error but the database continues to try to access the remote host and hangs.
6786222	When using Berkeley Database, sometimes the `sam-archiverd` hangs due to lock conflicts.
6757136	The archiver considers tape label time for choosing tapes only in some cases, not in all cases.
6911948	In a remote disk archive environment where the remote server is accessed via a WAN, single stream archive write performance drops dramatically when network latency increases.
6882254	Using `offline_copy` directly results in an archiver read timeout.
6876948	The `arcopy` process hangs when the disk archive volume is full rather than stopping and notifying the user.
6895723 and 6872668	Archiving stops, and the archiver trace file contains many "cache lookup failed" messages. Killing and restarting `sam-arfind` restarts archiving only temporarily, then it stops again.

Recycler

Bug Number	Description
6771843	Recycler unlinks archive files that are still writing to disk.
6788703	Running `sam-nrecycler` may result in loss of data.

6894132	The recycler hash table design causes performance problems, especially for large catalogs.
6872508	The recycler does not recover well from error and might still unlink files used by `arcopy`.
6872876	When hundreds of disk volumes are configured, recycler performance drops significantly.

Stager

Bug Number	Description
6827025	Staging fails and files are marked with damaged status, if the media is "STATUS_VOLUME_IN_USE."
6917209	When bad or non-existent volume or media parameters are used, `stagerd` goes into a loop, dumps a core file, and eventually dies.
6909454	`sam-stagealld` daemon dies without providing useful information to resolve the cause.
6888067	staging of files over 2GB in size fails
6886154	`sam-stagealld` dies and cannot be restarted by `sam-fsd`.
6903449	The `dio_min_size` threshold on the stager needs to be smaller (8 MB).
6834028	Occasionally, sam-stagerd_copy or sam-arcopy mark files as damaged when nothing is wrong with the files.
6862318	NFS file copy times can be quite long for offline SAM-QFS files.
6827689	`sam-stagerd_copy` always uses direct I/O even when user has explicitly disabled it.
6881720	Files that are migrated from a disk onto a SAM-QFS file system are marked "damaged" during staging.
6813177	For large numbers (thousands) of files, staging them from offline storage can be much slower than in the 4.6 release.

Configuration

6837405	SAM-QFS file system that was created on a ZFS pool appears to increase in size after unmounting and mounting.
6887901	SAM-QFS daemons fail to restart after a sequence of `umount`, `samfsgrow`, and `mount` processes.
6851383	Edits to clients section of `diskvols.conf` on remote server are not recognized by `samd config` or a reboot.

`samfsdump`

Bug Number	Description
6824033	If `samfsdump -uP` is used, files that have partial release set and no archive copies made have only the partial size dumped rather than the entire file.
6824036	If `samfsdump -P` is used and the partial size for the file system is large, files that are smaller than the partial size are dumped with the partial size. This causes `samfsrestore` to restore them with the partial size, and a subsequent `samfsck` detects excess data blocks.

Sideband Database

Bug Number	Description
6825691	For customers that use the sideband database, directory information becomes out of date in the database.
6825691	samdb update fails to update parent inode attributes.
6896876	Newly created files fail to get added to sideband database (samdb).
6838583	`samdb check` displays invalid error messages, although the consistency of the database might be affected.
6839852	On a new, empty file system (`sammkfs`), `samdb check <fsname> -s` dumps a core file.

6838468	inode checksum is not updated in the database.
6838124	If a consistency check is ran for a file that has multiple archive copies, copy 0 is not inserted into the archive table.
6838143	`samdb` consistency check does not update directory paths.

Storage Devices

Bug Number	Description
6905732	IBM LTO-4 WORM capability is broken.
6795522	`sam-genericd` dumps a core file and SAM-QFS hangs.
6812974	SAM-QFS tries to use a tape drive that is in the DOWN state.

Solaris Cluster and Oracle Real Applications Cluster (RAC)

Bug Number	Description
6687190	`samd buildmcf` does not support DID devices.
6611796	Shared QFS failover sometimes takes about 5 minutes to settle with the new server.
6560870	Shared QFS file system hangs during the sync of the file system during a panic.
6886903	In a COTC configuration, `mount_samfs` displays an error about a cross-device link when bringing a resource group online, although the file system is mounted.
6855718	Data corruption when NFS resource group primary node panics.
6855306	If a panic occurs while a failover to that node is in process, all file system clients go into a state from which you cannot easily recover because any subsequent failover to a different node will not be successful.
6850566	Software tries to failover to server that has "host off" set in the host table.
6843198	QFS file system does not mount at boot on some clients.
6714024	Locks held for dead QFS clients are not released.
6706459	A heavily loaded system displays many warning messages about "sam_putapage: sparse block..." errors.
6732953	When Solaris Cluster resources are disables and the cluster reboots, all nodes dump a core file and display `scqfs_boot` time out errors.
6866879	The `MC_CLUSTER_MGMT` flag is not always set when needed in SC configurations.
6838820	Shared QFS failover using SUNW.qfs agent should failover file systems in parallel instead of serially.
6828760	In an Oracle RAC configuration, when the cluster is shut down, messages are displayed that indicate the file system daemon is not responding.

Support Changes

Support Might Be Removed in a Future Release

The following might not be supported in a future product release:

- V1 inode and V1 superblock

Important User Clarifications

Excluding a Directory From **`samfsdump`**

When you want to exclude a directory, you must use a relative path with the -X excluded-dir option.

For example, the following command creates a dump file of everything in the /sam directory except for the files in the /sam/notme/notme2 directory.

```
example# cd /sam
example# samfsdump -f /dump-file-destination/samfsdump.today -X notme/notme2
```

Cleaning up Orphaned Blocks

Once in a while, a shrink operation might leave a device in NOALLOC state after the sam-shrink process exits. To resolve this issue, run the remove command again. If the device state still does not change to OFF, then there are "orphaned" blocks.

> Note -
> Depending on your environment, you might choose to leave the device in the NOALLOC state. NOALLOC means the device will not be used for additional allocation.

To clean up the orphaned blocks and change the state of the device to OFF, follow these steps:

1. Unmount the device.
2. Run samfsck -F to clean up the orphaned blocks.
3. Use either samu remove or samadm eq-remove to remove the device.

Shared File Systems Using Oracle RAC and Solaris Cluster

> Note -
> Do not mount shared file systems in the root (/) directory.

If you have configured your environment with Oracle Real Application Cluster (RAC) and Solaris Cluster and file systems are mounted in the root (/) directory, boot or failover operations might not operate as expected. Unrelated Solaris Cluster agents might interfere with your file system.

Known Limitations in This Release

Misleading Validation Message for Oracle RAC Configuration (CR 6810721)

When you configure a stand-alone Sun QFS file system for use with Oracle Real Application Clusters (RAC), you might see validation messages that are not appropriate.

If only the primary node has QFS standalone file systems mounted, the secondary node shows errors similar to the following when you tried to add the orasrv-rs resource:

```
clresource:  pocean2:global - Validation failed. ORACLE_HOME /orahome/product/10.2/db does not
exist
clresource:  pocean2:global - ALERT_LOG_FILE /orahome/product/10.2/db/rdbms/log/alert_swb.log
doesn't exist
clresource:  pocean2:global - PARAMETER_FILE: /orahome/product/10.2/db/dbs/initswb.ora nor
             server PARAMETER_FILE: /orahome/product/10.2/db/dbs/spfileswb.ora exists
clresource:  (C189917) VALIDATE on resource orasrv-rs, resource group OraDB, exited with non-zero
exit status.
clresource:  (C720144) Validation of resource orasrv-rs in resource group OraDB on node pocean2
failed.
clresource:  (C891200) Failed to create resource "orasrv-rs".
```

The steps that you followed to add the orasrv-rs resource are similar to the following:

```
pocean1:root>clresource create -g OraDB \
pocean1:root>-t SUNW.oracle_server -p ORACLE_HOME=/orahome/product/10.2/db \
pocean1:root>-p Alert_log_file=/orahome/product/10.2/db/rdbms/log/alert_swb.log \
pocean1:root>-p ORACLE_SID=swb -p Connect_string=oracle/dba1 \
pocean1:root>orasrv-rs
```

Workaround - Follow these steps:

1. Enable `fs-qfs-rs` on the primary node on which the file system is mounted.
2. On the secondary node, mount the file system using the command `mount -o reader` file-system-name.
3. Set a resource dependency between the `fs-qfs-rs` and `orasrv-rs` resources.

```
pocean1:root>clresource create -g OraDB \
pocean1:root>-t SUNW.oracle_server -p ORACLE_HOME=/orahome/product/10.2/db \
pocean1:root>-p Alert_log_file=/orahome/product/10.2/db/rdbms/log/alert_swb.log \
pocean1:root>-p ORACLE_SID=swb -p Connect_string=oracle/dba1 \
pocean1:root>-p resource_dependencies=fs-qfs-rs orasrv-rs
```

At this point resource creation will succeed, as indicated by the following example:

```
orasrv-rs           pocean1        Online        Online
                    pocean2        Offline       Offline
```

Related Topics

- What's New in This Release

What's New in This Release

Contents

What's New in the SAM-QFS 5.1 Release?

The SAM-QFS 5.1 release provides the foundation for the next generation of file system and storage archive management products. This release improves stability and performance over the previous SAM-QFS products. Although no significant new features are provided, several key enhancements are intended to ensure the safety of your data.

Specific fundamental improvements include the following:

- I/O path – Several bugs were fixed that relate to memory mapping I/O.
- Archiver operations – Several bugs were fixed that improve archiver performance and operations.
- Recycler operations – Several bugs were fixed that improve recycler performance and operations.
- Stager operations – Several bugs were fixed that improve stager performance and operations.
- Sideband database consistency – Several bugs were fixed that improve the consistency of the sideband database.
- Rolling upgrades – If you are already running SAM-QFS 5.0, you can upgrade individual shared file system clients to SAM-QFS 5.1 without taking down the rest of the file system. For more information, see Support for Rolling Upgrades in a Shared Environment.
- Improved remote disk archive performance, especially in a high-latency WAN environment – The remote disk archive write size on the client is now set to the value of the `blksize` parameter set in the `/etc/opt/SUNWsamfs/rft.cmd` file on the server. If no value is set, the default is 1 MB.
- Stager direct I/O behavior – The previous default for `dio_min_size` threshold on the stager was set at 32 MB. The threshold was changed to 8 MB to improve performance.

To view an abridged list by product feature of bugs fixed in this release, see Significant Bugs Fixed in This Release.

Also, a few key enhancements were made to the documentation for this release:

- Man pages are now viewable from the documentation wiki. See SAM-QFS Man Pages by Section.
- Information about using SAM-QFS with Solaris Cluster has been significantly enhanced. See Using SAM-QFS With Solaris Cluster.

Operating Systems

Sun QFS and SAM-QFS 5.0 software require the following minimum operating system release:

- Solaris 10, Update 6

In addition, you can use any of the following operating systems as a client in a shared file system:

- Solaris 10, Update 6 for x86 (32-bit)
- Red Hat Enterprise Linux 4.0 (UD-4) for x64 platforms
- Red Hat Enterprise Linux 4.5 for x64 platforms
- SuSE Linux Enterprise Server 9 (service pack 2) for x64 platforms
- SuSE Linux Enterprise Server 10 for x64 platforms
- SuSE Linux Enterprise Server 10 (service pack 2) for x64 platforms

Support for Rolling Upgrades in a Shared Environment

You can upgrade individual shared file system clients without taking down the rest of the file system. To perform a rolling upgrade, your environment must include one primary metadata server and at least one potential metadata server.

Follow these steps:

1. Upgrade the potential metadata server.
2. Failover to the potential metadata server.
3. Upgrade the primary metadata server.
4. Failover to the primary metadata server.
5. Upgrade the clients.

> Note -
> At any given time, the metadata server and the clients can only be one release apart.

Related Topics

- Sun QFS and SAM-QFS 5.1 Release Notes

Using a MySQL Database With SAM-QFS

Using a MySQL Database With SAM-QFS

About the Sideband Database

To improve performance for saving and restoring archive data, you can use the MySQL database. The `samdb`(1M) library reads event logs created by the `sam-fsalogd` daemon and issues SQL statements against the sideband database. Commands are executed through the `samdb` script. See the `samdb`(1M) man page for further documentation.

If the sideband database is enabled (`mount sam_db`), then the `sam-fsd` daemon starts up the file system activity log daemon (`sam-fsalogd`) when the file system is mounted. The `sam-fsalogd` daemon opens the event door, gets events from the file system, and logs the events to a log directory. The log directory can be specified in the `/etc/opt/SUNWsamfs/fsalogd.cmd` file.

Creating the Database

MySQL Requirements

The MySQL database does not have to be on the same system as the SAM-QFS software. For information about obtaining and installing MySQL, see http://www.mysql.com/.

Make sure that you have enough space on the MySQL server to store SAM-QFS information. Although exact numbers are difficult to specify, the following examples might help you to configure your disk space:

- 100,000,000 (One hundred million) files = 893 GB (less than a 1TB drive)
- 1,000,000,000 (One billion) files = 9TB

In one experience, 40 million files required 12 GB of storage for the database. Each file had two archive entries per file.

About the samdb.conf Configuration File

The `/etc/opt/SUNWsamfs/samdb.conf` file contains access parameters to the MySQL database for each family set. The following example shows entries for the samfs1 and samfs2 family sets.

```
samfs1:db.sun.com:3ksnn64:secret:samfs1:7009::/sam/s am1
samfs2:localhost:laura:secret:samfs2test:::/sam/sam2
```

For more information about the configuration file, see the `samdb.conf`(4) man page.

How to Enable Database Support

- To enable database support, use the `sam_db` mount parameter.
  For example:

  ```
  # mount -F samfs -o sam_db samfs1
  ```

  The `sam-fsd` daemon starts up a per file system activity log daemon `sam-fsalogd` as each file system is mounted. The `sam-fsalogd` daemon listens for events associated with its file system and logs them to files in the directory specified in the `fsalogd.cmd` file.

How to Initially Populate the Database

To populate the database, use the `samdb`(1M) load command with a load file created by `samfsdump`.

The following commands give examples of how the database load file can be created:

- To create a load file using current file system metadata, use `samfsdump`(1M).
  For example:

  ```
  # samfsdump -S -Z /tmp/samfs1/dbload /samfs1
  ```

  The `samfsdump -S` option creates a database load file and will prevent creation of a regular dump file.

- While restoring from an existing dump file, it may be useful to generate a database load file at the same time.
  For example:

  ```
  # samfsrestore -Z /tmp/samfs1/dbload -f /path/to/dump/samfs1.dump
  ```

- By adding the `-S` option to the command above, the same dump file can be used to generate a load file without actually restoring the files.
  For example:

  ```
  # samfsrestore -S -Z /tmp/samfs1/dbload -f /path/to/dump/samfs1.dump
  ```

- To directly populate the database with a single command using current file system metadata, use `samfsdump`(1M) and `samdb`(1M) in a

pipeline.
For example:

```
# samfsdump -S -Z - /samfs1 | samdb load samfs1
```

Using the Database

Improving Dump Performance

To improve the performance of `samfsdump`(1M), use the database for path name creation.
For example:

```
# samdb dump samfs1 | samfsdump -Y -f /path/samfs1.dump -
```

samdb Command Reference

The following is a brief list of available `samdb` sub-commands.

Sub-command	Description
check	Verifiy the information in the database against the inodes of the file system.
create	Create the database for the specified file system.
dump	Generate a list of files for `samfsdump`.
drop	Remove the file system from the database. You are asked to confirm removal.
load	Load information from a `samfsdump` file into the database. You should use the `check` sub-command to verify that the information is loaded correctly.
query	Query the database for information about a file or volume number.

For detailed information, see the `samdb`(1M) man page.

Resizing File Systems

Contents

Resizing File Systems

You can grow or shrink the size of a 5.0 or later file system without having to unmount the file system first. You can add a meta device, a data device, or a stripe group device. You can only remove a data device in an `ma` file system. There is no support for removing a device that has metadata (mm devices in an ma file system or md devices in an ms file system).

Note -

- Online grow and shrink only apply to 5.0 or later file systems with a version 2A superblock. You can convert existing file systems by using the `samfsck -u 2A` command. Once converted, these file systems are no longer compatible with versions of the software prior to 5.0.
- To grow older file systems, you must unmount the file system and then use the `samgrowfs` command. See Growing an Unmounted File System for more information.
- Online grow and shrink are not supported with Sun Cluster.

Shrinking a Mounted File System

About Online Shrink

Shrinking a file system involves two components:

- For SAM-QFS file systems, release marks files as OFFLINE and stops additional data from being allocated on the device.
    - If files are archived, the device is released almost immediately.
    - If files on the device have not yet been archived, the device is not moved into the OFF state.
    - If the release command fails to move the device into the OFF state, wait a sufficient time for files to be archived. Then, run the release command again to release any newly archived files. If there are files that cannot be archived, use the remove command to move the data that resides on the device to other available data devices.
- For QFS file systems, remove moves currently allocated blocks from the current device to an alternate device.
  During the removing process, the software performs these actions:
    1. When the remove is started, the equipment is put into the NOALLOC state to prevent any additional space being allocated on that device.
    2. The software starts a background process that copies files from the device to another device that is in the ON state. This step in the process might take some time to complete. To find out more about this process, see the `sam-shrink`(1M) man page.
    3. When the removing process is complete, the equipment is placed into the OFF state.

Important Notes About Online Shrink

- You can only shrink one device at a time. If a remove or release operation is in process, you must wait until it finishes before shrinking another device.
- You cannot remove the last ON device in a file system. In other words, you can't shrink the device to nothing.
- To remove a striped group:
    - Provide the equipment number of any device in the striped group.
    - You must have an equivalent striped group in the ON state to which to move the data. For example, if the existing striped group has four equipment numbers, you must have another striped group that is in the ON state and that has four equipment numbers.
- If you try to remove a device that contains metadata, the process fails and a detailed error message is written to the `/var/adm/messages` file.

How to Release a Device in a File System From the Command Line

When you release a device from a file system, the software frees all disk space for files that have a valid archive copy. You might want to perform this task to quickly release space on a device that is to be removed for hardware failure or other reasons.

Steps

1. To release the files on the disk, use one of the following commands:

```
# samcmd release <eq_number>
# samadm eq-release <eq_number>
```

2. Use the `samcmd m` utility to monitor the progress of the release process.

Example – Releasing a Device in a File System

```
# samadm eq-release 403
samadm: eq-release: eq release 403 started successfully.
Monitor 'samcmd m' display for completion and '/var/adm/messages' for errors.
```

How to Remove a Device in a File System From the Command Line

When you remove a device from a file system, any files that are on that device are moved to an alternate device and the removed device is marked as OFF.

1. To remove a device from a file system, use one of the following commands:

   ```
   # samcmd remove <eq_number>
   # samadm eq-remove <eq_number>
   ```

2. Use the `samcmd m` utility to monitor the progress of the remove process.

Example – Removing a Device in a File System

When the following example command executes:

1. LUN 404 is changed to the NOALLOC state.
2. The `sam-shrink` process starts moving the files to alternate LUNs that are in the ON state.
3. Once the data is completely moved, the state of LUN 404 is changed to OFF.

```
# samadm eq-remove 404
samadm: eq-remove: eq remove 404 started successfully.
Monitor 'samcmd m' display for completion and '/var/adm/messages' for errors.
```

How to Shrink a File System Using SAM-QFS Manager

> Note -
> You can only shrink a file system when the file system is mounted. You can only remove a data device in an ma file system. There is no support for removing a device that has metadata (mm devices in an ma file system or md devices in an ms file system).

1. From the Manage Hosts menu or the Managed Hosts page, choose the name of the server on which the file system is located.
   The File Systems Summary page is displayed.

2. Select the radio button next to the file system that you want to shrink.
   Make sure that the status of the file system is mounted as indicated by a disk usage value in the related Disk Usage (Total) column.

3. From the Operations menu, choose Shrink.
   The Shrink File System wizard is displayed.

4. Follow the steps in the Shrink File System wizard.
   For help in the wizard, click the wizard Help tab.

Growing a Mounted File System

About Online Grow

You can grow the size of a shared or unshared file system without having to unmount the file system first. You can add a metadata device, a data device, or a stripe group device. You can also replace equipment that is in the OFF state with new equipment.

How to Grow a File System From the Command Line

1. Add the new device to the `mcf` file.

   Note -
   Always add new devices at the end of the list of existing devices in the `mcf` file. Otherwise, the new devices do not show up and you cannot grow the file system.

2. Reload the configuration.

```
# samd config
```

3. Confirm.

```
# samcmd f
```

   The new device should show a state of OFF.

4. Add the new device to the existing file system.
   Use one of the following commands:

```
# samcmd add  <eq_number>
# samadm eq-add <eq_number>
```

5. Confirm.

```
# samcmd m
```

   The new device should show a state of ON. This device is now available to be used for allocating storage for the file system.

Example – Growing a Mounted File System

The following example adds device 404 to the current file system.

```
# samadm eq-add 404
samadm: eq-add: eq add 404 started successfully.
Monitor 'samcmd m' display for completion and '/var/adm/messages' for errors.
```

How to Grow a Shared File System From the Command Line

1. On the metadata server, add the new device to the `mcf` file.

   Note -
   Always add new devices at the end of the list of existing devices in the `mcf` file. Otherwise, the new devices do not show up and you cannot grow the file system.

2. On the metadata server, reload the configuration.

```
# samd config
```

3. Confirm.

```
# samcmd f
```

The new device should show a state of OFF.

4. On the metadata server, add the new device to the existing file system.
   Use one of the following commands:

```
# samcmd add <eq_number>
# samadm eq-add <eq_number>
```

5. Confirm that the device was added.

```
# samcmd m
```

   The new device should show a state of UNAVAIL.

6. On each client of the shared file system, add the new device to the `mcf` file.
   - For Solaris clients, you can update the `mcf` file while the client is mounted and running applications.

```
# samd buildmcf
# samd config
```

   - For Linux clients, you must umount the file system on each of the Linux clients, update the `mcf`, and then mount the client.

```
# umount
# samd config
# mount
```

   - For SANergy clients, you must unmount the file system, remount the file system, and refuse the file system on each client.

```
# umount
# mount
# SANergy config
```

7. On the metadata server, enable allocation for the updated file system.
   Use one of the following commands:

```
# samcmd alloc <eq-number>
# samadm eq-alloc <eq_number>
```

8. Confirm that the updated file system is ready to be used.

```
# samcmd m
```

   The new device should show a state of ON. This device is now available to be used for allocating storage for the file system.

How to Grow a File System Using SAM-QFS Manager

When you grow a file system, you increase storage capacity by adding devices to the file system.

1. From the Manage Hosts menu or the Managed Hosts page, choose the name of the server on which the file system is located.
   The File Systems Summary page is displayed.

2. Select the radio button next to the file system that you want to grow.

3. From the Operations menu, choose Grow.

The Grow File System wizard is displayed.

4. Follow the steps in the Grow File System wizard.
   For help in the wizard, click the wizard Help tab.

Growing an Unmounted File System

To increase the size of an unmounted file system, you add disk partitions or disk drives, and then update the `mcf` file and use the `samgrowfs` (1M) command to expand the file system. You do not need to reinitialize or restore the file system.

When making changes to the `mcf` file, be aware of the following:

- Add new partitions for metadata or data at the end of the `mcf` file, after the existing disk partitions.
- You must add both a data device and a metadata device.
- You can configure up to 252 disk partitions in a file system.
- To increase the size of a Sun QFS file system, you must add at least one new metadata partition. Metadata partitions require an Equipment Type value of `mm`.
- Do not change the Equipment Identifier name in the `mcf` file. If the name in the `mcf` file does not match the name in the superblock, the file system can no longer be mounted. Instead, the following message is logged in `/var/adm/messages`:

```
WARNING SAM-FS superblock equipment identifier <id> on eq <eq>
does not match <id> in mcf
```

How to Grow an Unmounted File System

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

```
# samd config
```

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

5. Issue the `samgrowfs`(1M) command on the file system that is being expanded.
   For example, type the following command to expand file system `samfs1`:

```
# samgrowfs samfs1
```

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

6. Mount the file system.
   For information about mounting a Sun QFS file system, see the `mount_samfs`(1M) man page.
7. Verify that the file system is ready to be used.

```
# samcmd m
```

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

```
# samd buildmcf
```

About QFS and SAM-QFS

Contents

About QFS and SAM-QFS

What is Sun QFS?

Sun™ QFS software is a high-performance file system that can be installed on Solaris x64 AMD and SPARC platforms. This high-availability file system ensures that data is available at device-rated speeds when requested by one or more users. The Sun QFS file system's inherent scalability enables the storage requirements of an organization to grow over time with virtually no limit to the amount of information that can be managed. This file system enables you to store many types of files (text, image, audio, video, and mixed media) all in one logical place. In addition, the Sun QFS file system enables you to implement disk quotas and a shared file system. This file system also includes the following features:

- Metadata separation
- Direct I/O capability
- Shared reader/writer capability
- File sharing in a storage area network (SAN) environment
- Sun Cluster support for high availability

What is Sun Storage Archive Manager (SAM-QFS)?

The Sun Storage Archive Manager (SAM-QFS) product enables you to archive file system data. The SAM-QFS environment includes a storage and archive manager along with Sun QFS file system software. The SAM-QFS software enables data to be archived to automated libraries at device-rated speeds. In addition, it enables data to be archived to files in another file system through a process known as disk archiving. You can archive the data on an "as-needed" basis, or you can define policies that determine when data should be archived. You can also set specific schedules for when to archive data. You are presented with a standard file system interface and can read and write files as though they were all on primary disk storage.

In a SAM-QFS configuration, the file system manages the online, nearline, and offline data automatically and in a manner that is transparent to the user or application. Users read and write files to the file system as though all files were on primary storage. In addition, this configuration backs up work in progress continually, automatically, and unobtrusively. Multiple file copies can be archived to many media types in a standard format. This minimizes the need for additional backup and provides fast disaster recovery in an effective long-term data storage solution.

The SAM-QFS software archives files by copying the files from online disk cache to archive media. The archive media can consist of disk slices in another file system, or it can consist of removable tape or magneto-optical cartridges in automated or manually loaded storage devices. In addition, the SAM-QFS software automatically maintains online disk space at site-specified usage thresholds. It releases disk space associated with archived file data and restores the files to online disk when they are needed.

The SAM-QFS configuration is especially suited to data-intensive applications that require a scalable and flexible storage solution, superior data protection, and fast disaster recovery. This configuration also includes the following features:

- Storage policy management
- Complete volume manager
- Disk-to-disk copying and archiving
- Shared tape drives
- Read-ahead/write-behind capability
- File segmentation

> Note -
> Although you can create and use Sun QFS file systems without the SAM-QFS archiving features, you cannot use the SAM-QFS archiving features without using Sun QFS file systems. As a result, you follow one of these installation paths:
>
> - Installing SAM-QFS (archiving and file system)
> - Installing Sun QFS (file system only)
>
> If you install just the file system and later decide that you want the archiving features, you must first uninstall Sun QFS then install SAM-QFS. Existing file systems are not deleted and do not need to be rebuilt.

What You Can Do With SAM-QFS

- Create file systems and configure them to be stand-alone, archiving, or shared
- Create file systems that are configured for use in high-performance computing environments
- Create stand-alone file systems
- Add clients and potential metadata servers to and remove them from shared file systems
- Grow file systems
- Shrink file systems
- Check and repair file systems
- Support alloc/noalloc on Logical Unit Numbers (LUNs)
- Mount and unmount stand-alone, shared, and archiving file systems
- Mount and unmount VERITAS File Systems (VxFS)
- Control archiving for archiving file systems
- Manage archive policies for archiving file systems
- Protect the data of mounted archiving file systems

What is the SAM-QFS Manager?

The SAM-QFS Manager software is a browser-based graphical user interface that enables you to configure, control, protect, and monitor one or more file systems in your network from a central location. To access this central location, you can use the web browser on any host in your network. The goal of the software is to provide a less complex way than the command line of performing the most common tasks associated with these file systems. To configure options that are unavailable through the browser interface, use the command-line interface and configuration files that are associated with the file systems.

SAM-Remote

The Sun SAM-Remote client and server storage management system enables you to share libraries and other removable media devices in a SAM-QFS environment. All host systems included in a Sun SAM-Remote environment must have the same SAM-QFS software release level installed and operational.

To configure the SAM-Remote software, create a file system that is configured for archiving. After the file system is tested and is known to be configured properly, you can use the SAM-Remote instructions to enable remote storage and archive management.

Using SAM-QFS for High Availability

High-Availability File System Configuration Using Sun Cluster (HA-QFS)

You can install a Sun QFS file system in a Sun Cluster environment and configure the file system for high availability. The following configuration methods are available, depending on whether your file system is shared or unshared:

- In a shared file system, when the Sun Cluster software fails over, it moves the Sun QFS file system operations from the failing server to a different server. The Sun Cluster software then moves the metadata server's operations from a failing node to another node without requiring you to enter further commands.
- You can also have shared clients outside of the cluster in a Sun Cluster environment.
- An unshared Sun QFS file system configured in a Sun Cluster environment is a highly available file system. Such a file system is configured on one node but is enabled as a highly available resource within the cluster. When the node hosting the file system fails, the Sun Cluster software moves the file system to another node.

For more information about these configurations, see Using SAM-QFS With Solaris Cluster.

> Note -
> Although installing a Sun QFS file system in a Sun Cluster environment improves reliability and decreases or eliminates unplanned downtime, it does not eliminate planned downtime. In order to maintain the health of the file system, the Sun QFS software may need to be brought down occasionally to run the `samfsck` process. It also needs to be shut down in order to apply software patches or updates.

High-Availability Archiving Configuration Using Sun Cluster (HA-SAM)

SAM-QFS can be configured for high availability by using Sun Cluster software. Sun Cluster software provides high availability by enabling application failover. The primary node is periodically monitored and the cluster software automatically relocates the SAM-QFS archiving functions from a failed primary node to a designated secondary node. By allowing another node in a cluster to automatically host the archiving workload when the primary node fails, Sun Cluster software can significantly reduce downtime and increase productivity.

High-availability SAM-QFS (HA-SAM) depends on the Sun QFS Sun Cluster agent, so this configuration must be installed with a shared Sun QFS file system that is mounted and managed by the Sun QFS Sun Cluster agent.

For more information, see Configuring Archiving in a Solaris Cluster Environment (HA-SAM).

About Shared File Systems and the Linux Client

A shared file system is a distributed, multihost file system that you can mount on multiple Solaris Operating System (Solaris OS) hosts. One Solaris OS host acts as the metadata server, and the others act as clients. You can also designate one or more clients as potential metadata servers, enabling you to switch metadata servers.

Within a shared file system, the Sun QFS software can be installed on Linux clients as well as on Solaris clients. Unlike a shared Sun QFS Solaris client, the Linux client is restricted to client-only behavior. It cannot be configured as a potential metadata server. The Linux client supports interaction with SAM-QFS software, but has Sun QFS file system functionality only.

The Sun QFS software functionality is largely the same for the Solaris and Linux clients. For more information about the Sun QFS Linux client software, see Using SAM-QFS on Linux Clients.

Related Topics

- More About SAM-QFS
- More About QFS
- Sun QFS and SAM-QFS 5.1 Release Notes
- New Features in SAM-QFS 5.1

Next Steps

- Planning Your Environment
- Installing SAM-QFS
- Installing Just QFS
- Installing SAM-QFS Manager

Backing Up SAM-QFS Data and Files

Contents

Backing Up SAM-QFS Data and Files

This section describes the recommended procedures for regularly backing up important data and files in the Sun SAM-QFS environment.

Backing Up File System Data

This section describes the recommended procedures for regularly backing up important data and files in a QFS file system-only environment. For information about backing up in an archiving environment. see Creating Archive Recovery Points.

Setting Up Dump Files

File systems are made up of directories, files, and links. The file system keeps track of all the files in the `.inodes` file. The `.inodes` file resides on the metadata device. The file system writes all file data to the data devices. Depending on your configuration, the metadata device can be separate from the file system data device.

It is important to use the `qfsdump`(1M) command periodically to create a dump file of metadata and file data. The dump process saves the relative path information for each file contained in a complete file system or in a portion of a file system. This protects your data in the event of a disaster.

You can create dump files as often as once or twice a day, depending on your site's requirements. By dumping file system data on a regular basis, you can restore old files and file systems. You can also move files and file systems from one server to another.

Follow these guidelines for creating dump files:

- The `qfsdump`(1M) command dumps file names, inode information, and data. This command creates full dumps, not incremental dumps, of specified files and directories, and the resulting file can therefore be very large. The `qfsdump`(1M) command does not have any tape management, size estimations, or incremental dump facilities, as does `ufsdump`(1M). In addition, the `qfsdump`(1M) command does not support volume overflow, so you need to weigh space considerations and make sure that the size of the file system does not exceed the size of the dump media.
- The `qfsdump`(1M) command dumps all the data of a sparse file, and the `qfsrestore`(1M) command restores all the data. These commands do not, however, preserve file qualities that enable sparse files to be characterized as sparse. This can cause files to occupy more space on dump files and on restored file systems than anticipated.
- Because you issue the `qfsdump`(1M) command on a mounted file system, inconsistencies can arise as new files are being created on disk. Dumping file systems during a quiet period (a time when files are not being created or modified) is a good idea and minimizes these inconsistencies.
- Ensure that you dump metadata and data for all file systems. Look in `/etc/vfstab` for all file systems of type `samfs`.

You can run the `qfsdump`(1M) command manually or automatically. Even if you implement this command to be run automatically, you might also need to run it manually from time to time, depending on your site's circumstances. In the event of a disaster, you can use the `qfsrestore`(1M) command to re-create your file system. You can also restore a single directory or file.

For more information about creating dump files and restoring from them, see the `qfsdump`(1M) man page. The following sections describe procedures for issuing this command both manually and automatically.

How to Run the **`qfsdump`**(1M) Command Automatically Using **`cron`**

Perform this step for each file system in your environment. Make sure you save each dump file in a separate file.

1. For each file system, make an entry in the root `crontab` file so that the `cron` daemon runs the `qfsdump`(1M) command periodically.

For example:

```
10 0 * * * (cd /qfs1; /opt/SUNWsamfs/sbin/qfsdump -f /dev/rmt/0cbn)
```

This entry executes the `qfsdump`(1M) command at 10 minutes after midnight. It uses the `cd`(1) command to change to the mount point of the `qfs1` file system, and it executes the `/opt/SUNWsamfs/sbin/qfsdump` command to write the data to tape device `/dev/rmt/0cbn`.

How to Run the **qfsdump**(1M) Command Manually From the Command Line

1. Use the `cd`(1) command to go to the directory that contains the mount point for the file system.
   For example:

   ```
   # cd /qfs1
   ```

2. Use the `qfsdump`(1M) command to write a dump file to a file system outside of the one you are dumping.
   For example:

   ```
   # qfsdump -f /save/qfs1/dump_file
   ```

Creating Archive Recovery Points

About Recovery Points

You can use the information stored in a recovery point to recover an archiving file system in the event of a disaster. It is important to use SAM-QFS Manager or the `samfsdump`(1M) command periodically to create these recovery points.

When using the `samfsdump`(1M) command, note the following:

- The `samfsdump`(1M) command dumps file names and inode information, not data. That is, the dump file does not include the archive data stored in your file system. The dump file does include the inode and directory structure information necessary to quickly locate the data on your archive media. This information is necessary for recovering from a file system failure. For more information, see the `samfsdump`(1M) man page.
- You can use the `-u` option to the `samfsdump`(1M) command to back up metadata and file data for files that have not yet been archived. A `samfsdump`(1M) snapshot taken using the `-u` option can be very large. Unlike `ufsdump`(1M), the `samfsdump`(1M) command does not have any tape management or estimation capability. You need to weigh the trade-offs of space and unarchived data when using the `-u` option. For more information about these commands, see the `samfsdump`(1M) and `ufsdump`(1M) man pages.
- If a failure occurs after file system initialization, you can use File System Manager or the `samfsrestore`(1M) command to restore data using the dump file.
  For more information about using the `samfsdump`(1M) command, see the `samfsdump`(1M) man page. Also see the information on metadata, disaster preparation, and recovery in the SAM-QFS Troubleshooting.

How to Create a Recovery Point (**samfsdump**(1M) Command) From the Command Line

1. Use the `cd`(1) command to go to the directory that contains the mount point for the file system.
   For example:

   ```
   # cd /samfs1
   ```

2. Use the `samfsdump`(1M) command to write the output to a file system outside of the one that you are backing up.
   For example:

```
# samfsdump -T -u -f /dumpster/dump.file
```

If you have a specific list of files that you would like to dump, you can use the `-I` include_file option.This only dumps the files and directories that are listed in the include_file. The include_file must have one relative or absolute path per line.

How to Schedule the **`samfsdump`**(1M) Command Using **`cron`**

1. Make an entry in the root user's `crontab` file so that the `cron` daemon runs the `samfsdump`(1M) command periodically.
   The following code example shows a cron(1) entry.

```
0 0 * * * find /csd.directory/sam -type f -mtime +7 \
-print| xargs -l1 rm -f; cd /sam; \
/opt/SUNWsamfs/sbin/samfsdump- f \
/csd.directory/sam/&rsquo;date +\%y\%m\%d&rsquo;
```

This example `crontab` entry uses a QFS file system mounted on `/sam`. Replace `/csd.directory` with an existing directory of your choice. This entry causes the commands to execute each day at midnight. First, the old dumps are renamed and a new dump is created in `/csd.directory/sam/`yymmdd. After that, `cron`(1M) emails the `samfsdump`(1M) output to `root`.Troubleshooting

If you have multiple QFS file systems, make similar `crontab` entries for each file system. Save each dump in a separate file.

How to Create a Recovery Point Using SAM-QFS Manager

You can create a recovery point from the SAM-QFS Manager interface at any time.

Follow these steps to create a recovery point:

1. From the Servers menu, select the server on which the file system that you want to administer is located.
   The File Systems Summary page is displayed.
2. Navigate to the Recovery Points node under File Browsing & Recovery in the navigation tree.
3. Select the file system for which you want to create a recovery point in the drop down menu.
4. Click the Create Recovery Point Now... button.
   The Take Recovery Point pop-up window is displayed.
5. In the Fully Qualified Recovery Point Name field, type the path and the name of the recovery point file that you want to create.
6. Click Submit.
   For more information on creating recovery points, see the SAM-QFS Manager online help.

How to Schedule Automatic Recovery Points Using SAM-QFS Manager

Scheduling a recovery point through the SAM-QFS Manager interface is the equivalent of creating a `crontab`(1) entry that automates the Sun SAM-QFS software process.

Follow these steps to schedule a recovery point:

1. From the Servers menu, select the server on which the archiving file system that you want to administer is located.
   The File Systems Summary page is displayed.
2. Select the radio button next to the archiving file system for which you want to schedule a recovery point.
3. From the Operations menu, choose Schedule Recovery Points.
   The Schedule Recovery Points page is displayed.
4. Specify values on the Schedule Recovery Points page.
   For instructions on using this page, see the SAM-QFS Manager online help.
5. Click Save.

Backing Up Configuration Files

The software regularly accesses several files that are created during the installation and configuration procedure. You should back up these files regularly to a file system that is outside the file system in which they reside. In the event of a disaster, you can then restore these files from your backup copies.

You should back up the following files regularly and whenever you modify them:

- /etc/opt/SUNWsamfs/mcf
- /etc/opt/SUNWsamfs/samfs.cmd
- /etc/opt/SUNWsamfs/defaults.conf
- /etc/opt/SUNWsamfs/archiver.cmd

For more information about the files you should protect, see SAM-QFS Troubleshooting.

Related Topics

Next Steps

Installing SAM-QFS Software

Installing SAM-QFS Software

The Sun QFS and Sun Storage Archive Manager (SAM-QFS) products are closely linked. Depending on the features that you need, choose from the following:

Feature	Packages	Required License	For More Information
Archiving (local file system)*	SUNWsamfsr, SUNWsamfsu	SAM-QFS	Installing and Configuring SAM-QFS
Archiving (shared file system)	SUNWsamfsr, SUNWsamfsu	SAM-QFS and QFS	Installing and Configuring SAM-QFS
Local or shared file system (no archiving)	SUNWqfsr, SUNWqfsu	QFS	Installing Sun QFS
Browser-based management	SUNWfsmgrr, SUNWfsmgru	SAM-QFS or QFS	Use fsmgr_setup as explained in Installing SAM-QFS Manager.

* Formerly referred to as SAM-FS.

Installation Process

- Quick Start - Installing Sun QFS and SAM-QFS
- Installing and Configuring SAM-QFS
  - Planning Your Environment
  - Release Package Contents
  - Preparing for Installation
  - Installing the Software Packages
  - Configuring the File System Environment
  - Initializing the Environment
  - Setting Up Mount Parameters
- Installing Sun QFS
- Performing Additional SAM-QFS Configuration
- Installing SAM-QFS Manager
- Uninstalling SAM-QFS Manager
- Upgrading Hardware
- Upgrading QFS and SAM-QFS
- Upgrading the Solaris OS
- Complete (Printable) SAM-QFS 5.1 Installation Guide
- Complete (Printable) Sun QFS 5.1 Installation Guide

Next Steps

- Configuring the File System
- Configuring a Shared File System
- Configuring Storage Devices for Archiving
- Configuring the Archiver

Complete (Printable) SAM-QFS 5.1 Installation Guide

Quick Start - Installing Sun QFS and SAM-QFS

Contents

Installing Sun QFS and SAM-QFS - Quick Start

The following instructions are a high-level overview of the installation process. For more detailed instructions, see Detailed Installation Instructions for SAM-QFS and Detailed Installation Instructions for Just QFS.

Before You Begin

- If you are not familiar with the SAM-QFS product, see About QFS and SAM-QFS.
- If this is a new installation or you have made significant changes to your storage and file system environments, see Planning Your Environment.
- Check the hardware and software requirements in Preparing for Installation.

How to Install the Packages

1. Insert the installation CD into your computer or go to the directory into which you downloaded the packages.
2. Select the correct package for your environment.
   For example:

   ```
   cd sparc
   ```

   Or:

   ```
   cd x64
   ```

3. Select the correct Solaris version.

   ```
   cd 2.10
   ```

4. Install the packages.
   - To install the SAM-QFS (archiving and file system) functionality, type the following command:

     ```
     pkgadd -d . SUNWsamfsr SUNWsamfsu
     ```

   - To install just the file system functionality, type the following command:

     ```
     pkgadd -d . SUNWqfsr SUNWqfsu
     ```

     Note -
     To use the WORM-FS package for Write-Once, Read-Many (WORM) functionality, you must also install the `SUNWsamfswm` package.

5. Verify the installation.
   - To verify a SAM-QFS installation, type the following command:

     ```
     pkginfo |grep sampkginfo -l SUNWsamfsr
     ```

   - To verify a QFS-only installation, type the following command:

     ```
     pkginfo |grep qfspkginfo -l SUNWqfsr
     ```

6. Run `sync`.

How to Enable SAM-QFS Logging

1. Edit the `/etc/syslog.conf` file by adding the following lines:

   ```
   # SAM-QFS logging
   local7.debug /var/adm/sam-log
   ```

   Note -
   Use a tab (as opposed to a space) between between the file names.

2. Issue the following commands:

   ```
   touch /var/adm/sam-log
   pkill -HUP syslogd
   ```

How to Enable SAM-QFS Daemon Tracing

1. Copy the example `defaults.conf` file into the correct location.
   For example:

   ```
   # cp -i /opt/SUNWsamfs/examples/defaults.conf /etc/opt/SUNWsamfs
   ```

2. Edit the `/etc/opt/SUNWsamfs/defaults.conf` file to enable tracing.
   Information in the `defaults.conf` file should be similar to the following:

```
trace
all = on
endtrace
```

How to Set Up the root PATH Variables

- Edit the `/.profile` file to set up the PATH variables:

```
PATH=$PATH:/opt/SUNWsamfs/bin:/opt/SUNWsamfs/sbin:/opt/SUNWsamfs/tools
MANPATH=/opt/SUNWsamfs/man:/opt/SUNWsan/man:$MANPATH
export PATH MANPATH
```

How to Set Up Stand-alone QFS

1. Create the `mcf` file.
   a. Copy an example `mcf` file.

```
cd /etc/opt/SUNWsamfs
cp -i /opt/SUNWsamfs/examples/mcf .
```

   b. Edit the `mcf` file to be similar to the following example:

```
#
# Equipment       Equip.  Equip.  Family  Device  Addit.
# Identifier      Number  Type    Set     State   Params.
# --------------- -----  ------  ------- -------  -------
qfs1              10     msqfs1-
/dev/dsk/c5t16d0s011mdqfs1-
/dev/dsk/c5t17d0s012mdqfs1-
/dev/dsk/c5t18d0s013mdqfs1-
/dev/dsk/c5t19d0s014mdqfs1-
/dev/dsk/c5t20d0s015mdqfs1-
/dev/dsk/c5t21d0s016mdqfs1-
```

2. Create the `/etc/vfstab` file.
   For example:

```
#DEVICE    DEVICE   MOUNT   FS    FSCK  MOUNT    MOUNT
#TO MOUNT  TO FSCK  POINT   TYPE  PASS  AT BOOT  PARAMETERS
#
qfs1       -        /qfs1   samfs  -    yes
```

3. Create the mount point and mount the file system as shown in the following example series of commands:

```
mkdir /qfs1
chmod 755 /qfs1
samd config
sammkfs qfs1
mount qfs1
chmod 777 /qfs1
```

How to Set Up Shared QFS

1. Create the `mcf` file.

For example:

```
# Equipment        Eq Eq Family Dev Additional
# Identifier       Nm Tp Set    St  Parameters
# ---------------- -- -- ------ --- ----------
#
sqfs1                            10 ma qfs3 - shared
/dev/dsk/c2t50020F2300000C98d0s2 11 mm qfs3 -
/dev/dsk/c2t50020F2300004921d0s2 12 mm qfs3 -
/dev/dsk/c2t50020F2300000C98d0s3 13 mr qfs3 -
/dev/dsk/c2t50020F2300004921d0s3 14 mr qfs3 -
#
sqfs2                                             20 mb qfs4 - shared
/dev/dsk/c2t50020F2300000C98d0s4                  21 mm qfs4 -
/dev/osd/osd0100080020E1381F00002A00479F7D98,root 22 o1 qfs4 -
/dev/osd/osd010000144F94A14C00002A00479F4EB4,root 23 o1 qfs4 -
```

Note -
For the ms file system type in Shared QFS, set `stripe=2` either in `/etc/vfstab` or `samfs.cmd` so that metadata is distributed across all devices.

2. Create the hosts files.
   a. Copy the example hosts file.

   ```
   cp -i /opt/SUNWsamfs/examples/hosts* .
   ```

   b. Create the hosts file for each file system.
   A hosts file is required for each file system. You must include a server priority for any server that will be a metadata server.
   The following example is for the `hosts.sqfs1` file.

   ```
   #
   # Host Name  Network Interface  Server Priority Unused Server
   # ------------------------------------------------------------
   kingkong    kingkong-priv,kingkong   1      -     server
   godzilla    godzilla-priv,godzilla   2      -
   nebula      nebula-priv,nebula       -      -
   cosmic      cosmic-priv,cosmic       -      -
   ```

   The following example is for the `hosts.sqfs2` file.

   ```
   #
   # Host Name  Network Interface   Server Priority Unused    Server
   # ----------------------------------------------------------------
   walleye     walleye-priv0,walleye      1       -          server
   bass        bass-priv0,bass            2       -
   northern    northern-priv0,northern    -       -
   muskie      muskie-priv0,muskie        -       -
   mallard     mallard-priv0,mallard      -       -
   wood        wood-priv0,wood            -       -
   ruddy       ruddy-priv0,ruddy          -       -
   mandarin    mandarin-priv0,mandarin    -       -
   ```

3. Create the `vfstab` file, similar to the following example.

```
#device  devicemountFSfsckmount    mount
#to mount to fsckpointtypepassat bootoptions
#
sqfs1     -/sqfs1samfs-noshared
sqfs2     -/sqfs2samfs-yesshared,stripe=2,bg
```

4. Create the mount points and mount the file systems:
   a. Create the mount points and set file permissions on the mount points.

      Note -
      You must create the mount points and set appropriate permissions on all hosts that are part of the shared file system.

```
mkdir /sqfs1
chmod 755 /sqfs1

mkdir /sqfs2
chmod 755 /sqfs2
```

   b. Verify that SAM-QFS is correctly configured in the `/etc/inet/services` file.
      For example, use the `tail` command as shown in the following example:

```
tail /etc/inet/services
```

      The result of the `tail` command shows the appropriate SAM-QFS service, similar to the following example:

```
sam-qfs   7105/tcp# SAM-QFS
```

   c. Reconfigure the `sam-fsd` daemon based on the new `mcf` file and related configuration changes.

```
samd config
```

   d. Create the shared file systems using the mount points that you defined above.

```
sammkfs -S sqfs1
sammkfs -S sqfs2
```

      Tip -
      The -S option identifies the file system as a shared file system.

   e. Mount the new file system on the metadata server.

```
server# mount sqfs1
server# mount sqfs2
```

   f. Mount the new file systems on the clients.

```
client# mount sqfs1
client# mount sqfs2
```

   g. Confirm that space has been allocated for the file systems.

```
df -1h
```

5. Run the following if the configuration changes:

```
samd config
```

How to Enable Network Time Protocol Daemon (Shared QFS)

1. Edit the `/etc/inet/ntp.conf` ">file:

```
server nettime prefer
server earth
```

2. Issue the following commands:

```
#sync
#reboot
```

How to Set Up the Network File System

1. Set up the network file system (NFS) server.
   Edit the `/etc/dfs/dfstab` file to include information similar to the following example:

```
share /sqfs1
```

2. Set up the NFS client.
   Edit the `/etc/vfstab` to include information similar to the following example:

```
# NFS - 300 second timeout needed for failover
kingkong:/sqfs1 -/nssqfs1nfs-notimeo=3000
```

How to Install the SAM-QFS Manager

To use the browser-based interface to create and manage file systems and archiving, follow these steps.

1. Go to the top level of the installation directory.
   For example: `/net/mymachine/packages-to-load/SAM-QFS_5.0/SUN_QFS_5.0/sparc`
2. Run `fsmgr_setup`.

Commands to Monitor a Shared File System

Use the following commands to monitor a shared file system. For detailed information about the commands and their options, see the applicable man pages.

```
samfsinfo sqfs1
samsharefs sqfs1
samcmd N sqfs1
```

You can also use the `samu` operator utility, which is especially useful when you have more than one file system.

Related Topics

- Detailed Installation Instructions for SAM-QFS
- Detailed Installation Instructions for Just QFS

Next Steps

- Sun Storage Archive Manager (SAM-QFS) Configuration and Administration Guide
- Sun QFS File System Configuration and Administration Guide
- Using SAM-QFS With Solaris Cluster

Installing and Configuring SAM-QFS

Contents

Complete (Printable) SAM-QFS 5.1 Installation Guide

Installing and Configuring SAM-QFS

The Sun QFS and Sun Storage Archive Manager (SAM-QFS) products are closely linked. Depending on the features that you need, choose from the following:

Feature	Packages	Required License	For More Information
Archiving (local file system)*	`SUNWsamfsr`, `SUNWsamfsu`	SAM-QFS	Read this document.
Archiving (shared file system)	`SUNWsamfsr`, `SUNWsamfsu`	SAM-QFS and QFS	Read this document.
Local or shared file system (no archiving)	`SUNWqfsr`, `SUNWqfsu`	QFS	See Installing Sun QFS.
Browser-based management	`SUNWfsmgrr`, `SUNWfsmgru`	SAM-QFS or QFS	Use `fsmgr_setup` as explained in Installing SAM-QFS Manager.

* Formerly referred to as SAM-FS.

Complete the tasks below if this is the initial installation of the SAM-QFS (archiving and file system) software packages at your site.

> **Note**
> You must be logged in as superuser to complete the installation tasks.

Before You Begin

- If you are not already familiar with Sun QFS and SAM-QFS, see About QFS and SAM-QFS.
- Before you follow the detailed installation steps below, check the hardware and software requirements as explained in Preparing for Installation.

Installation Overview Task Map

Depending on the the features that you need to support, you must complete several of the following procedures.

Step	Task	Description
1	Add the software packages.	Install the appropriate packages for your needs. Also see Release Package Contents.
2	Configure `path` and `manpath` variables.	Configure the environment variables for access to commands and man pages.
3	(Optional) Install and configure SAM-QFS Manager.	This task is needed only if you want to use a browser to configure file systems and archiving features.
4	(Optional) Configure tape and magneto-optical storage devices.	This task is needed only if you want to use tape or magneto-optical devices for archiving.
5	Configure the file system environment.	Define the master configuration file `mcf`.
6	Configure file system mount parameters.	Define the `/etc/vfstab` and `samfs.cmd` files.
7	Initialize the environment.	Initialize SAM-QFS and mount the file systems.
8	Configure file archiving.	Define parameters for archiving file systems to media.

Installing the Software Packages

The Sun QFS and Sun Storage Archive Manager (SAM-QFS) software uses the Sun Solaris packaging utilities for adding and deleting software. The `pkgadd`(1M) utility prompts you to confirm various actions necessary to install the packages.

Table – Release Package Names

Installed Package	Description
`SUNWqfsr` `SUNWqfsu`	Sun QFS (file system only) software packages
`SUNWsamfsr` `SUNWsamfsu`	SAM-QFS(archiving and file system) software packages

How to Add the Packages

1. Become superuser.
2. Use the `cd` command to go to the directory where the software package release files reside.
   You obtained the release files as described in Obtaining the Release Files. Changing to the appropriate directory differs, depending on your release media, as follows:
   - If you downloaded the release files, go to the directory to which you downloaded the files.
   - If you obtained the release files from a CD-ROM, go to the directory on the CD-ROM that corresponds to your operating system version.
3. Use the `pkgadd`(1M) command to add the appropriate packages, based on the features that you need to support:
   - For archiving to a local or shared file system, install the `SUNWsamfsr` and `SUNWsamfsu` packages.

- To support just a local or shared file system (no archiving), install the SUNWqfsr and SUNWqfsu packages.
  For example:

```
# pkgadd -d . SUNWsamfsr SUNWsamfsu
```

4. When prompted to define an administrator group, select yes or y to accept the default (no administrator group), or select no or n if you want to define an administrator group.
   You can reset permissions on certain commands later by using the set_admin(1M) command. For more information about this command, see Adding the Administrator Group or the set_admin(1M) man page.
5. Examine the SAM-QFS installation log file /tmp/SAM_install.log.
   This file should show that the pkgadd(1M) command added the SUNWsamfsr and SUNWsamfsu software packages. Verify that the SAM-QFS samst driver is also installed. If all files installed properly, the following message appears:

```
Restarting the sysevent daemon
```

How to Set Up PATH and MANPATH Variables

To enable access to the commands and man pages for the Sun QFS and SAM-QFS commands, you must modify your PATH and MANPATH environment variables.

1. For users who will need to access the user commands, such as sls(1), add /opt/SUNWsamfs/bin to the users' PATH variables.
2. Edit your system setup files to include the correct paths to commands and man pages.
   - In the Bourne or Korn shells, change the PATH and MANPATH variables in the .profile file and export the variables.
     The following example shows how your .profile file might look after editing.

```
PATH=$PATH:/opt/SUNWsamfs/bin:/opt/SUNWsamfs/sbin
MANPATH=$MANPATH:/opt/SUNWsamfs/man
export PATH MANPATH
```

   - In the C shell, edit the .login and .cshrc files.
     The path statement in your .cshrc file might look like the following example:

```
set path = ($path /opt/SUNWsamfs/bin /opt/SUNWsamfs/sbin)
```

     The MANPATH statement in your .login file might look like the following example:

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

Installing and Configuring SAM-QFS Manager

If you want to use the browser user interface to configure and manage your SAM-QFS environment, see Installing SAM-QFS Manager.

Configuring Tape and Magneto-Optical Storage Devices

If you need to enable archiving to tape or magneto-optical media, follow the instructions in Configuring Storage Devices for Archiving. You do not need to perform these tasks if you plan to archive to disk.

Configuring the File System Environment

Each SAM-QFS software environment is unique. The system requirements and hardware differ from site to site. SAM-QFS environments support a wide variety of tape and optical devices, automated libraries, and disk drives. The system administrator at your site must set up the specific configuration for your environment.

The master configuration file, `/etc/opt/SUNWsamfs/mcf`, defines the equipment topology managed by the SAM-QFS software. This file specifies the devices, automated libraries, and file systems included in the environment. You assign each piece of equipment a unique Equipment Identifier in the `mcf` file.

You can edit the `mcf` file in either of two ways:

- By using the SAM-QFS Manager interface to configure archiving and file system devices. When you create a file system using SAM-QFS Manager, it creates an `mcf` file in `/etc/opt/SUNWsamfs/mcf` that contains a line for each device and family set of the file system.

- By directly editing the `mcf` file using a text editor.

The `mcf` file has two kinds of entries:

- File system device entries for disk devices. In the `mcf` file, you organize them into one or more file systems.
- Removable media device entries that you can organize into family sets. The `mcf` file contains information that enables you to identify the drives to be used and associate them with the automated libraries to which they are attached.

For detailed information about `mcf` file structures and contents, see About the Master Configuration File.

Example `mcf` files are installed in `/opt/SUNWsamfs/examples`. Several example `mcf` file configurations are also provided in Examples of `mcf` Files.

The following sections provide examples and describe activities related to creating and maintaining the `mcf` file.

How to Create the Master Configuration File (mcf) Manually

- Use vi(1) or another editor to create the `/etc/opt/SUNWsamfs/mcf` file.
  For detailed information about the contents of the `mcf` file, see About the Master Configuration File.

  > You can copy an example `mcf` file from `/opt/SUNWsamfs/examples` or from the examples in Examples of `mcf` Files.

  When you create the `mcf` file, follow these guidelines:

  - Delimit the fields in each line with spaces or tabs.
  - Begin each comment line entered into this file with a pound sign (#).
  - Use a dash (-) to indicate optional fields that are omitted.
    The following example shows the `mcf` file fields.

```
#
# Sun  Storage Archive Manager file system configuration
#
# Equipment       Equip  Equip Fam   Dev    Additional
# Identifier      Ord    Type  Set   State  Parameters
# ----------      -----  ----- ----  -----  ----------
```

    The `mcf` file can contain both comment lines and lines that pertain to a device. The types of lines that can pertain to a device are as follows:

  - Family set parent identifiers and family set devices
  - Family set member devices
  - Stand-alone devices

Identifying Peripherals Using the /var/adm/messages File

When your system boots, a series of messages is written to `/var/adm/messages`. These messages identify the Sun Solaris hardware path to each of the peripherals on your system. You can use this information to create the `mcf` file. To display information from the latest system reboot, search backward from the end of the file.

As the following example shows, each SCSI peripheral has three lines. The sixth field, `samst2`, indicates that these lines are associated with each other.

Example – SCSI Peripheral Lines in the **/var/adm/messages** File

```
# tail -200 /var/adm/messages | more
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@5,8800000/samst@2,0
```

- The first line displays the vendor and product information that the SCSI peripheral reported to the Sun Solaris kernel.
- The second line displays the SCSI bus, SCSI target ID, and LUN of the peripheral.
- The third line displays the peripheral's hardware path. This path is reflected in the `/devices` directory. Symbolic links (symlinks) to the `/devices` directory are set up in the `/dev/st`, `/dev/samst`, and `/dev/rmt` directories. Note that the third line wraps to the next line.

Matching the symbolic link to the correct peripheral is the key to configuring a Sun Storage Archive Manager environment. Use the `ls`(1) command with the `-l` option in both the `/dev/st`, `/dev/samst` and `/dev/rmt` directories to display the path name of the peripheral.

You might also want to set up the "device down" notification script at this point. The `dev_down.sh`(1M) man page contains information about setting up this script, which sends email to root when a device is marked down or `off`. For more information, see the `dev_down.sh`(1M) man page.

How to Verify the mcf File

- If you created your `mcf` file manually, use the `sam-fsd`(1M) command to verify the file.
  If you created your `mcf` file using SAM-QFS Manager, you do not need to verify its syntax.
  If the `mcf` file is free of syntax errors, the `sam-fsd`(1M) output includes information about the file systems, archiving, and other system information. If the `mcf` file contains syntax or other errors, however, the output is similar to the following example.

```
# sam-fsd
13: /dev/dsk/c1t1d0s0   10      md      samfs1  on      /dev/rdsk/c1t1d0s0
*** Error in line 13: Equipment name '/dev/dsk/c1t1d0s0' already in use by eq 10
72: /dev/rmt/3cbn       45      ug      l1000   on
*** Error in line 72: Equipment name '/dev/rmt/3cbn' already in use by eq 44
2 errors in '/etc/opt/SUNWsamfs/mcf'
sam-fsd: Read mcf /etc/opt/SUNWsamfs/mcf failed.
```

  If the `mcf` file has errors, return to Setting Up the Environment Configuration and refer to the `mcf`(4) man page for information about creating this file. You can also refer to Examples of `mcf` Files.

How to Create an mcf File Using SAM-QFS Manager

Before You Begin

When you configure QFS file systems using the SAM-QFS Manager software, it creates or edits the appropriate configuration files, including the `mcf` file, on that server. You can use either SAM-QFS Manager or the CLI to further edit these files later.

> Note -
> If you want to use SAM-QFS Manager to configure your archiving environment and you want to include network attached libraries (excluding STK Libraries) in this configuration, you must create your parameters file before you create the `mcf` file. For information about creating a parameters file, see Creating Parameters Files for Network Attached Automated Libraries.
> You can add a StorageTek ACSLS network library in the SAM-QFS Manager without creating the parameters file. The application automatically generates the parameters file for you when you add the library in the Library Summary Page.

Steps

1. From your web browser, log in to the SAM-QFS Manager as an administrator user.
2. Expand the Getting Started section and choose First Time Configuration.
3. In section 2, click Create a File System.

The New File System wizard is displayed.

4. Follow the steps for creating a new file system.
   When you have completed this process, the `mcf` file is created. For more information, see the SAM-QFS Manager online help.

Setting Up Mount Parameters

Use the procedures in this section to specify mount parameters for the file system.

You can specify mount parameters in the following ways:

- Using the `mount`(1M) command. Mount options specified here override those specified in the `/etc/vfstab` file and in the `samfs.cmd` file.
- In the `/etc/vfstab` file. Mount options specified here override those specified in the `samfs.cmd` file.
- In the `samfs.cmd` file.

For a list of available mount options, see the `mount_samfs`(1M) man page.

Updating the /etc/vfstab File and Creating the Mount Point

This section describes how to edit the `/etc/vfstab` file.

The following table shows the values you can provide in the fields in the `/etc/vfstab` file.

Table – **`/etc/vfstab`** File Fields

Field	Field Title and Content
1	Device to Mount. The name of the file system to be mounted. This value must be the same as the file system's Family Set name specified in the `mcf` file.
2	Device to `fsck`(1M). Must be a dash (`-`) character, which indicates that there are no options. This character prevents the Solaris system from performing an `fsck`(1M) process on the file system. For more information about this process, see the `fsck`(1M) or `samfsck`(1M) man page.
3	Mount Point, for example, `/samfs1`.
4	File System Type. Must be `samfs`.
5	`fsck`(1M) Pass. Must be a dash (`-`) character, which indicates that there are no options.
6	Mount at Boot. Either yes or no. • Specifying `yes` in this field indicates that the Sun Storage Archive Manager file system is to be mounted automatically at boot time. • Specifying `no` in this field indicates that you do not want to mount the file system automatically. For information about the format of these entries, see the `mount_samfs`(1M) man page.
7	Mount Parameters. A list of comma-separated parameters (with no spaces) that are used in mounting the file system. You can specify mount options on the `mount`(1M) command, in the `/etc/vfstab` file, or in a `samfs.cmd` file. Mount options specified on the `mount`(1M) command override those specified in the `/etc/vfstab` file and in the `samfs.cmd` file. Mount options specified in the `/etc/vfstab` file override those in the `samfs.cmd` file. For a list of available mount options, see the `mount_samfs`(1M) man page.

When you create a file system using SAM-QFS Manager, a default `/etc/vfstab` file is created. However, mount options specified in SAM-QFS Manager are written to the `samfs.cmd` file rather than to the `/etc/vfstab` file. For more information, see Creating and Editing the `samfs.cmd` File.

To edit the mount options in the `/etc/vfstab` file, use the following procedure, How to Update the /etc/vfstab File and Create the Mount Point Using a Text Editor.

How to Update the /etc/vfstab File and Create the Mount Point Using a Text Editor

The example in this task assumes that `/samfs1` is the mount point of the `samfs1` file system.

1. In the `/etc/vfstab` file, create an entry for each file system.

The following example shows header fields and entries for a local file system.

```
#DEVICE    DEVICE   MOUNT   FS    FSCK  MOUNT    MOUNT
#TO MOUNT  TO FSCK  POINT   TYPE  PASS  AT BOOT  PARAMETERS
#
samfs1     -        /samfs1 samfs  -    yes      high=80,low=60
```

2. Use the mkdir(1M) command to create the mount point.
   For example:

```
# mkdir /samfs1
```

Creating and Editing the samfs.cmd File

You can create the /etc/opt/SUNWsamfs/samfs.cmd file as the place from which the system reads mount parameters. If you are configuring multiple file systems with multiple mount parameters, consider creating this file.

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

How to Create and Edit the samfs.cmd File Using SAM-QFS Manager

If you specify non-default mount options when creating a file system in SAM-QFS Manager, the samfs.cmd file is created or updated with those mount options.

Follow these steps to edit a file system's mount options:

1. On the Managed Hosts page, click the name of the server on which the file system is located.
   The File Systems Summary page is displayed.
2. Select the radio button next to the file system whose mount options you want to edit.
3. From the Operations menu, choose Edit Mount Options.
   The Edit Mount Options page is displayed.
4. Make your edits in the fields.
   For more information about the fields on the Edit Mount Options page, see the File System Manager online help.
5. Click Save.
   The new mount options are written to the samfs.cmd file.

How to Create and Edit the samfs.cmd File Using a Text Editor

- Use vi(1) or another editor to create the samfs.cmd file.
  Create lines in the samfs.cmd file to control mounting, performance features, or other aspects of file system management. For more information about the samfs.cmd file, see The samfs.cmd File or the samfs.cmd(4) man page.

Initializing the Environment

This section tells you how to initialize the environment and the file system, and how to mount the file system.

How to Initialize the Environment

1. Use the samd(1M) config command to initialize the archiving and file system environment.
   For example:

```
# samd config
```

How to Initialize the File System

This procedure describes how to use the sammkfs(1M) command and the Family Set names that you have defined to initialize a file system.

Note -
The sammkfs(1M) command sets one tuning parameter, the disk allocation unit (DAU). You cannot reset this parameter without reinitializing the file system. For information about how the DAU affects tuning, see File Allocation Methods and the sammkfs(1M) man page.

1. Use the sammkfs(1M) command to initialize a file system for each Family Set name defined in the mcf file.

   Caution -
   Running the sammkfs(1M) command creates a new file system. It removes all references to the data currently contained in the partitions associated with the file system in the /etc/opt/SUNWsamfs/mcf file.

   The following example shows the command to initialize a file system with the Family Set name of samfs1.

```
# sammkfs samfs1
sammkfs: Configuring file system
Building "samfs1" will destroy the contents of devices:
    /dev/dsk/c2t0d0s3
    /dev/dsk/c2t0d0s7
Do you wish to continue? [y/N] *y*
total data kilobytes       = 16777728
total data kilobytes free  = 16777152
#
```

   The actual numbers returned vary from file system to file system.

Mounting the File System

The mount(1M) command mounts a file system and reads the /etc/vfstab and samfs.cmd configuration files. For information, see the mount_samfs(1M) man page.

How to Mount the File System Using SAM-QFS Manager

1. On the Managed Hosts page, click the name of the server on which the file system is located.
   The File Systems Summary page is displayed.
2. Select the radio button next to the file system that you want to mount.
3. From the Operations menu, choose Mount.

How to Mount the File System From the Command Line

1. Use the mount(1M) command to mount the file system.
   Specify the file system mount point as the argument. For example:

```
# mount /samfs1
```

   Tip -
   If the file system has not been added to the /etc/vfstab file, use the following form of the mount command:

```
#mount -F samfs <fs_name> </mount_point>
```

2. Use the `mount(1M)` command with no arguments to verify the mount.
This step confirms that the file system is mounted and shows how to set permissions. The following example shows the output from a `mount(1M)` command issued to verify whether example file system `samfs1` is mounted.

```
# mount
_<<< information deleted >>>_
/samfs1 on samfs1 read/write/setuid/intr/largefiles/onerror=panic/dev=8001e3 on Thu Feb  5
11:01:23 2004
_<<< information deleted >>>_
```

3. (Optional) Issue the `chmod(1)` and `chown(1)` commands to change the permissions and ownership of the file system's `root` directory. Typically, this step is performed if this is the first time that the file system has been mounted. For example:

```
# chmod 755 /samfs1
# chown root:other /samfs1
```

Next Steps

- Performing Additional SAM-QFS Configuration
- Installing SAM-QFS Manager

Planning Your Environment

Contents

Planning Your Environment

If you are installing Sun QFS or Sun Storage Archive Manager (SAM-QFS) for the first time, you should make some decisions before you actually install the software. This section describes some of the guidelines that you should follow and decisions that you should make.

What Features Do You Need?

The following table lists the features that the Sun QFS and SAM-QFS products provide and provides links to additional considerations for each feature.

Feature	For More Information	Guidelines
Copy files (archiving)	Archiving Overview	Design Considerations for Archiving
Share archive media remotely	SAM-Remote	Design Considerations for Archiving
Enable failover for archiving	High-Availability Configuration Using Sun Cluster	Using SAM-QFS With Solaris Cluster
Manage files without archiving	File System Overview	Design Considerations for File Systems Without Archiving
Enable failover for non-archiving file systems	High-Availability Configuration Using Sun Cluster	Using SAM-QFS With Solaris Cluster

Archiving Overview

The Sun Storage Archive Manager (SAM-QFS) software archives files by copying the files from online disk cache to archive media. The archive media can consist of disk slices in another file system, or it can consist of removable tape or magneto-optical cartridges in automated or manually loaded storage devices. In addition, the Sun SAM software automatically maintains online disk space at site-specified usage thresholds. It releases disk space associated with archived file data and restores the files to online disk when they are needed.

High-Availability Configuration Using Sun Cluster

SAM-QFS can be configured for high availability by using Sun™ Cluster software. Sun Cluster software provides high availability by enabling application failover. The primary node is periodically monitored and the cluster software automatically relocates the archiving functions from a failed primary node to a designated secondary node. By allowing another node in a cluster to automatically host the archiving workload when the primary node fails, Sun Cluster software can significantly reduce downtime and increase productivity.

High-availability SAM-QFS (HA-SAM) depends on the SAM-QFS Sun Cluster agent, so this configuration must be installed with a shared Sun QFS file system that is mounted and managed by the Sun Cluster agent for SAM-QFS.

For more information, see Using SAM-QFS With Solaris Cluster.

SAM-Remote

The Sun SAM-Remote client and server storage management system enables you to share libraries and other removable media devices in a SAM-QFS environment. All host systems included in a Sun SAM-Remote environment must be running the same SAM-QFS software release level.

To configure the SAM-Remote software:

1. Create a Sun QFS file system.
2. Test the Sun QFS file system to verify that it is configured properly.
3. Use the SAM-Remote instructions to enable remote storage and archive management.

For more information, see Using the Sun SAM-Remote Software.

Design Considerations for Archiving

The following are some of the design considerations that need to be taken into account in the planning and implementation of an archiving environment:

- Access and usage patterns will have a significant impact on the tape systems required, including the types of tapes and the number and types of drives and libraries needed. If much staging is anticipated, linear rather than helical scan tapes are strongly recommended.
- The following will strongly influence the number of drives and media required:
  - File system sizes and usage patterns: frequency of file updates, average file sizes, batch jobs that will wait for drives/media or users that must have priority
  - Archiving policies: number of media copies, release policies, level of disk over-subscription
  - Media characteristics: size, drive setup times
    For best performance, Fibre Channel tape drives and disk devices should be accessed through separate host bus adapters (HBAs).

If you are managing a server that has the SAM-QFS software installed locally and you are configuring stand-alone file systems on the server to be archiving, it is recommended that you have at least one tape library associated with the current server. The library must contain media of a single media type.

The following table describes archiving configuration guidelines, on a per-tape-library basis, that can prevent you from overextending your environment.

Table – Archiving Configuration Guidelines

Number of Tape Drives	Number of Archive Policies (Sets)	Max. Number of File Systems	Max. Number of Files per File System	Library Recycler Values

2–3	1	4	6 million	• Minimum Gain - 90 • VSN Limit (#) - 2 • High-Water Mark - 50 • Size Limit - 30 Gbytes
4–5	1	6	6 million	• Minimum Gain - 90 • VSN Limit (#) - 3 • High-Water Mark - 50 • Size Limit - 40 Gbytes
6–7	2	10	8 million	• Minimum Gain - 90 • VSN Limit (#) - 5 • High-Water Mark - 50 • Size Limit - 50 Gbytes
8–10	4	10	10 million	• Minimum Gain - 90 • VSN Limit (#) - 8 • High-Water Mark - 50 • Size Limit - 70 Gbytes

> Note -
> The number of file systems in a configuration relates directly to the hardware purchased for the usage of the file systems. For example, your ability to support millions of file systems depends on having the right hardware (CPUs, memory, storage devices, and so on).

Your customer requirements drive the maximum number of files in a file system. On an average system, you should be able to restore 100 million files in less than 24 hours. If you do not need to restore files in 24 hours, youi can have more files in the file system.

The following are some further considerations that can help you from overloading your archiving system:

- Tape drives are designed to write large amounts of data at one time, so a well-designed archiving system should reduce the number of loads for the tape drives and increase the amount of data being written at one time.
- If you have only one tape drive with one media type, the `startage`, `startsize`, and `startcount` archive parameters should be set as follows:
  - `startage`– no less than 8 hours
  - `startsize`– no less than 50% of the capacity of a single tape
  - `startcount`– use a number in the thousands; do not exceed 500,000
    Do not run the recycler more than three times per day.

Design Considerations for File Systems Without Archiving

The Sun QFS software requires a certain amount of disk cache (file system devices) to create and manage data files and directories. An ma-type file system requires at least two disk devices or partitions, one for file data and one for metadata. An ms-type file system only requires one partition, on which both data and metadata are saved. Multiple disk devices or partitions increase I/O performance. See File System Design Basics for a detailed description of the file system types.

The disk devices or partitions do not require any special formatting. You might see better performance if you configure multiple devices across multiple interfaces (HBAs) and disk controllers.

> Caution -
> Make sure that the disks and partitions that you plan to use are not currently in use and do not contain any existing data. Any existing data will be lost when you create the file system.

The disks must be connected to the server through a Fibre Channel or SCSI controller. You can specify individual disk partitions for a disk, or you can use the entire disk as a disk cache. The software supports disk arrays, including those under the control of volume management software, such as Solaris Volume Manager, and other volume management software products.

Before creating your first file system, you should familiarize yourself with file system layout possibilities. For information on volume management, file system layout, and other aspects of file system design, see File System Overview.

Note -
Extensible Firmware Interface (EFI) labels are required on all shared disks if you are using a shared file system configuration that contains both the Solaris 10 OS on x64 platforms and the Solaris 10 OS on SPARC platforms. See Configuring EFI Labels for Shared x64 and SPARC Volumes for information on relabeling disks.

Reference Architecture

Use the following configuration recommendations to have fewer tape hardware and tape media problems. These recommendations also miminize the time that the archiver uses the tape drives. This frees the table drives to be used more for staging files.

The hardware is the primary factor that drives the configuration. Most storage environments are similar to the following:

- One Tape library
- One Media type
- Ten or less tape drives (Most sites have four)

Based on this hardware configuration, use the following global parameters:

- If disk archiving is in place, use the following settings:

```
allsets -sort path -offline_copy stageahead  -reserve set
allsets.1  -startage 10m  -startsize 500M -startcount 500000 -drives 6 -archmax  1G
allsets.2  -startage 24h -startsize 20G -startcount 500000 -drives X  -archmax 24G
allsets.3  -startage 48h -startsize 20G -startcount 500000 -drives X  -archmax 24G
```

Note -
Although it is not required that you use disk archiving, you should use disk archiving. Because the tape media is getting larger, disk archiving protects the data while the data accumulates to provide larger write operations (20GBytes for each write operation).
If you cannot wait as long as eight hours before a file is written to tape, you should use disk archiving.

- If disk archiving is not in place, use the following settings:

```
allsets -sort path -offline_copy stageahead  -reserve set
allsets.1  -startage 8h    -startsize 8G   -startcount 500000 -drives X -archmax  10G
allsets.2  -startage 24h -startsize 20G -startcount 500000 -drives X  -archmax 24G
```

The `releaser.cmd` file should be changed to:

```
list_size = 300000
```

The `stager.cmd` file should have these values:

```
maxactive  =  500000  # If server has more than 8G of RAM
maxactive  =  100000  # If server has less than 8G of RAM
```

Best Practices

- Most customers should use the `ms` file system configuration.
- The segment size set on the disk storage should be 512K.
- The storage should be configured with RAID 5 3+1 or 4+1 (no virtual volumes).
- The `ms` file system runs faster when you have more dedicated data LUNs and HBAs. The number of server I/O slots have an impact on this recommendation.

Next Steps

- Installing SAM and QFS Together
- Installing Just QFS
- Installing SAM-QFS Manager

Release Package Contents

Contents

Release Package Contents, Directories, and Files

Release Package Contents

The Sun Storage Archive Manager (SAM-QFS) and Sun QFS software packages are in Sun Solaris `pkgadd(1M)` format. These packages reflect the Sun Solaris version for the platform on which you will be installing the software.

The following table shows the release package names.

Table – Release Package Names

Installed Package	Description
`SUNWqfsr` `SUNWqfsu`	Sun QFS (file system only) software packages
`SUNWsamfsr` `SUNWsamfsu`	SAM-QFS (archiving and file system) software packages
`SUNWfsmgrr` `SUNWfsmgru`	SAM-QFS Manager software packages; Run the `fsmgr_setup` script to install
`SUNWsamfswm`	WORM-FS support packages

The releases are identified using characters arranged in the following format:
major U update . patch

The U stands for "update" in this format.

In the patch number field, a number between 1 and 99 indicates a patch release, and a letter from A through Z indicates pre-release software. The base release of a first feature release of a major release might not contain a patch level.

For example:

- 4U0 is release 4, update 0, a major release with no minor release revisions and no bug fixes.
- 4U2 is release 4, update 2, a minor release.
- 4U2.1 is a patch release that contains software fixes for a major or minor release. This number appears in the patch's README file.

Directories and Files Created

This section describes the directories and files associated with the Sun QFS and SAM-QFS products. You can obtain additional information about the files in this section from the man pages after the software is installed.

Directories Created at Installation

The following table lists the directories created when the software packages are installed.

Table – Directories Created

Directory	Content
`/dev/samst`	Device driver special files (only when SAM-QFS packages are installed).
`/etc/fs/samfs`	Commands specific to the software.
`/etc/opt/SUNWsamfs`	Configuration files.
`/etc/opt/SUNWsamfs/scripts`	Site-customizable scripts.
`/opt/SUNWsamfs/bin`	User command binaries.
`/opt/SUNWsamfs/client`	Files for remote procedure call API client.
`/opt/SUNWsamfs/doc`	Documentation repository for any informational files included in the release. The README file, which summarizes the installed release's features, is included in this directory.
`/opt/SUNWsamfs/examples`	Various example configuration files.
`/opt/SUNWsamfs/include`	API include files.
`/opt/SUNWsamfs/lib`	Relocatable libraries.
`/opt/SUNWsamfs/man`	Man pages.
`/var/snmp/mib`	Standard MIB files and product MIB (`SUN-SAM-MIB.mib`).
`/opt/SUNWsamfs/sbin`	System administrator commands and daemon binaries.
`/opt/SUNWsamfs/sc`	Sun Cluster binaries and configuration files.
`/opt/SUNWfsmgr/bin`	SAM-QFS Manager administrator commands.
`/opt/SUNWfsmgr/doc`	SAM-QFS Manager online documentation repository.
`/var/opt/SUNWsamfs`	Device catalogs, catalog trace file, log files, and archiver data directory and queue files.

Files Created at Installation

The following table lists miscellaneous files created when the software is installed.

Table – Files Created - Miscellaneous

File	Description
`/etc/opt/SUNWsamfs/inquiry.conf`	Vendor and product identification strings for recognized SCSI devices (only when SAM-QFS packages are installed).
`/etc/sysevent/config/SUNW,SUNWsamfs,sysevent.conf`	Solaris system event handler configuration file.
`/kernel/drv/amd64/samaio`	File system asynchronous I/O pseudo-driver (64-bit version for x64 platforms).
`/kernel/drv/amd64/samioc`	Sun Solaris 64-bit file system interface module (for x64 platforms).
`/kernel/drv/amd64/samst`	SAM-QFS driver for SCSI media changers and optical drives for tape drives (64-bit version for x64 platforms).
`/kernel/drv/samaio.conf`	Configuration file for `samaio`.
`/kernel/drv/samioc.conf`	Configuration file for the `samioc` module.
`/kernel/drv/samst.conf`	Configuration file for the `samst` driver.

/kernel/drv/sparcv9/samaio	File system asynchronous I/O pseudo-driver (64-bit version for SPARC platforms).
/kernel/drv/sparcv9/samioc	Sun Solaris 64-bit file system interface module (for SPARC platforms).
/kernel/drv/sparcv9/samst	SAM-QFS driver for SCSI media changers and optical drives for tape drives (64-bit version for SPARC platforms).
/kernel/fs/amd64/samfs	Sun Solaris 64-bit file system module for the x64 platform.
/kernel/fs/sparcv9/samfs	Sun Solaris 64-bit file system module for SPARC platforms.
/var/log/webconsole/host.conf	SAM-QFS Manager configuration file.
/var/opt/SUNWsamfs/faults	Faults history file.
/var/sadm/samqfsui/fsmgr_uninstall	Software for removing SAM-QFS Manager and its supporting applications.
/opt/SUNWsamfs/sc/etc/SUNW.qfs	Sun Cluster configuration file created only in the presence of Sun Cluster software.
/usr/cluster/lib/rgm/rtreg/SUNW.qfs	Sun Cluster configuration file created only in the presence of Sun Cluster software.

The file system has dynamically loadable components that are stored in the Sun Solaris `/kernel` directory. You can use `modinfo`(1M) command to determine the modules that are loaded. Typically, the kernel loads the file system module at boot time. Alternatively, you can load the file system module when the file system is first mounted after the Sun software is installed.

Fault Notification Files

After the software is installed, it creates files that it uses for fault notification. The following table lists these files. When the software detects faults serious enough to merit user attention, the software uses these trap and log files to convey fault information through the SAM-QFS Manager software.

Table – Files Created - Fault Notification

File	Description
/etc/opt/SUNWsamfs/scripts/sendtrap	Sends trap information.
/opt/SUNWsamfs/sbin/fault_log	Records faults.
/opt/SUNWsamfs/sbin/tapealert_log	Records `tapealert`(1M) faults (only when SAM-QFS packages are installed).
/opt/SUNWsamfs/sbin/tapealert_trap	Sends `tapealert`(1M) traps (only when SAM-QFS packages are installed).

The software creates these files with the following permissions:

```
-rwxr-x---
```

Caution -
Do not change these file permissions.
If execute permissions are lost, for example, the system writes messages such as the following to `/var/adm/messages`:

```
SUNW,SUNWsamfs,sysevent.conf, line1: no execute access to
/opt/SUNWsamfs/sbin/tapealert_trap - No such file or directory.
```

Site Files

The configuration procedures elsewhere in this information direct you to create several site-specific files. The software uses these site files.

> **Note** -
> Your site's configuration files must contain ASCII characters only.

You must create the master configuration `mcf` file at your site in order to use the Sun SAM-QFS software. For more information about the `/etc/opt/SUNWsamfs/mcf` file, see About the Master Configuration File and the `mcf`(4) man page.

If you are using the archiver and file system features, you might also create all the files shown in the following table. If you are using only file system features, you might only create the first two files.

Table – Optional Site Files

File	Description
`/etc/opt/SUNWsamfs/samfs.cmd`	File system mount parameter command file. For more information, see the `samfs.cmd`(4) man page.
`/etc/opt/SUNWsamfs/defaults.conf`	Miscellaneous default values. For more information, see the `defaults.conf`(4) man page.
`/etc/opt/SUNWsamfs/archiver.cmd`	Archiver command file. For more information, see the `archiver.cmd`(4) man page, or Configuring the Archiver.
`/etc/opt/SUNWsamfs/preview.cmd`	Previewer command file. For more information, see Configuring the Stager and the `preview.cmd`(4) man page.
`/etc/opt/SUNWsamfs/recycler.cmd`	Recycler command file. For more information, see Configuring the Recycler and the `recycler.cmd`(4) man page.
`/etc/opt/SUNWsamfs/releaser.cmd`	Releaser command file. For more information, see About Releasing and the `releaser.cmd`(4) man page.

Modified System Files

During installation, the software adds information to certain Solaris system files. These system files are ASCII text files. The Solaris OS uses these files to identify loadable kernel modules by number rather than by name.

The software adds information to the following files:

- `/etc/name_to_major`
  The SAM-QFS software uses this file to map drivers to major numbers. The `samst` and `samrd` major numbers can vary, depending on the major numbers in use by the Solaris OS. The system adds the following lines to this file:

```
samst 63
samrd 64
samioc 236
samaio 237
```

- `/etc/security/auth_attr`
  This file is the authorization description database. The system adds the following lines to this file:

```
# File System Manager Authorizations
com.sun.netstorage.fsmgr.config:::File System Manager All Access::
com.sun.netstorage.fsmgr.operator.media:::File System Manager Media
Related Operation Access::
com.sun.netstorage.fsmgr.operator.sam.control:::File System Manager
Start/Stop/Idle Archiving Access::
com.sun.netstorage.fsmgr.operator.file:::File System Manager File
LevelOperation Access::
com.sun.netstorage.fsmgr.operator.filesystem:::File System Manager
FileSystem Level Operation Access::
```

- `/etc/user_attr`
  This file is the extended user attributes database used by SAM-QFS Manager.

```
root::::profiles=Web Console Management,All;auths=
Solaris.*,solaris.grant,*com.sun.netstorage.fsmgr.**;
lock_after_retries=no
```

- `/etc/inittab`
  The system adds the following line to this file:

```
sfad:3:respawn:/opt/SUNWsamfs/sbin/fsmgmtd
```

Preparing for Installation

Contents

Preparing for Installation

This section explains the system requirements for the Sun QFS and SAM-QFS products and the tasks you must complete before you begin to install and configure your software.

Hardware and Software Requirements

You can install the software either on a Sun server based on UltraSPARC ® technology or on a server based on AMD Opteron x64 technology.

Additional requirements for the server that you want to use as the web server host for the SAM-QFS Manager browser interface tool are described in SAM-QFS Manager Requirements.

The software package runs on many Sun workstations and servers. Before installation, you should verify the compatibility of the hardware and the level of the Solaris Operating System (OS).

Operating System Requirements

Before installation, verify the applicability of the hardware and the level of the operating system. To install the software, you also must have root-level access to your system.

Sun Storage Archive Manager and Sun QFS 5.1 software require the following minimum operating system release:

- Solaris 10, Update 6

In addition, you can use any of the following operating systems as a client in a shared file system:

- Solaris 10 OS for x86 (32-bit)
- Red Hat Enterprise Linux 4.0 (UD-4) for x64 platforms
- Red Hat Enterprise Linux 4.5 for x64 platforms
- SuSE Linux Enterprise Server 9 (service pack 2) for x64 platforms
- SuSE Linux Enterprise Server 10 for x64 platforms
- SuSE Linux Enterprise Server 10 (service pack 2) for x64 platforms

How to Verify the Environment

Repeat these steps for each host on which you want to install the software.

1. Verify that your system has a CD-ROM drive or that it can access the release package at the Sun Download Center
2. Log in to your system as `root`.
   You must have superuser access to install the software.
3. Verify your system's Solaris OS level.
   For example, the output from the following command will show the major and minor OS release information as well as the architecture:

   ```
   % cat /etc/release
   ```

   The software relies on properly configured Solaris software at the following minimum release level:

   - Solaris 10, Update 6

Installing Solaris OS Patches

Sun Microsystems provides Solaris OS patches to customers with a maintenance contract by means of CD-ROM, anonymous FTP, and the Sun Microsystems SunSolve™ web site .

To install a patch after you install the Sun QFS or SAM-QFS release packages, load 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 included in the patch or jumbo patch cluster.

Software Host Requirements

If you plan to install the software in a multihost environment, such as a SAM-Remote configuration, all host systems must have the same software release level installed and operational.

Verifying Shared File System Requirements

This section describes the system requirements for a Sun QFS shared file system.

Metadata Server Requirement

You must have at least one Solaris metadata server. If you want to be able to change the metadata server, you must have at least one other Solaris host that can become the metadata server. These additional host systems are known as potential metadata servers. These servers must all be running on the same hardware platform, either SPARC or x64. You cannot mix server hardware platforms. In a Sun Cluster environment, all nodes included in a shared file system are potential metadata servers.

The following are configuration recommendations with regard to metadata storage:

- A shared file system should have multiple metadata (`mm`) partitions. This spreads out metadata I/O and improves file system throughput.
- A shared file system should use a separate, private metadata network so that typical user traffic does not interfere with metadata traffic. A switch-based (not hub-based) network is recommended.

Operating System and Hardware Requirements

Ensure that your configuration meets the following operating system and hardware requirements:

- The host systems to be configured in the Sun QFS shared file system must be connected by a network.
- All metadata servers and potential metadata servers must have the same processor type.
- The client systems can be installed on the Solaris OS or on one of the following operating systems:
  - Red Hat Enterprise Linux 4.0 (UD-4) for x64 platforms (Sun QFS shared client only)
  - Red Hat Enterprise Linux 4.5 for x64 platforms (Sun QFS shared client only)
  - SuSE Linux Enterprise Server 9 (service pack 2) for x64 platforms (Sun QFS shared client only)
  - SuSE Linux Enterprise Server 10 for x64 platforms (Sun QFS shared client only)
  - SuSE Linux Enterprise Server 10 (service pack 2) for x64 platforms (Sun QFS shared client only)
- Online data storage devices must be directly accessible to all hosts. All online metadata storage devices must be directly accessible to all potential metadata server hosts.

Sun Storage Archive Manager and Sun QFS Release Levels

Ensure that your configuration meets the following requirements:

- Each host to be configured in the shared file system must have the same software package installed.
- All software installed on the systems in the shared file system must be at the same release level. For example, if one host has the SAM-QFS 5.1 packages, all hosts that are part of the shared file system must have the SAM-QFS 5.1 packages.
  This requirement ensures that all systems in a shared file system have identical over-the-wire protocol versions. If these levels do not match, the system writes the following message to the metadata server's `/var/adm/messages` file when mounting is attempted:

  ```
  SAM-FS: <client> client package version <x> mismatch, should be <y>.
  ```

- When applying patches or upgrading the software for a shared file system, make sure to apply the same patch to all hosts that have access to the shared file system. Unexpected results might occur if not all host systems are running the same patch level.

Verifying Third-Party Compatibilities

The SAM-QFS software interoperates with many different hardware and software products from third-party vendors. Depending on your environment, you might need to upgrade other software or firmware before installing the SAM-QFS package. Consult the Release Notes for information pertaining to library model numbers, firmware levels, and other compatibility information.

SAM-QFS Manager Requirements

The SAM-QFS Manager browser interface is used to configure, control, monitor, or reconfigure a SAM-QFS environment using a graphical web browser interface.

You can install the SAM-QFS Manager software in one of the following configurations:

- As a stand-alone management station to manage one or more Sun QFS hosts
- As additional software on the Sun QFS host

After the SAM-QFS Manager software is installed, you can invoke the SAM-QFS Manager from any machine on the network that is allowed access to its web server.

For information about the requirements for the host upon which you are configuring the SAM-QFS Manager software, see Verifying Requirements for SAM-QFS Manager.

Determining Disk Space Requirements

The SAM-QFS software packages require a certain amount of disk cache (file system devices) to create and manage data files and directories.

Planning Your File System and Verifying Disk Cache

A local file system requires only a single partition. If you install SAM-QFS to enable archiving support, the file system requires either one or two partitions:

- To store file data separately from file system metadata (`ma` file system), you need to have at least two disk devices or partitions.
- To store data and metadata on the same device (`ms` file system), you need to have one disk device or partition.

The disk devices or partitions do not require any special formatting. You might see better performance if you configure multiple devices across multiple interfaces (HBAs) and disk controllers.

> Caution -
> Make sure that the disks and partitions that you plan to use are not currently in use and do not contain any existing data. Any existing data will be lost when you create the Sun QFS file system.

The disks must be connected to the server through a Fibre Channel (FC) or SCSI controller. You can specify individual disk partitions for a disk, or you can use the entire disk as a disk cache. The software supports disk arrays, including those under the control of volume management software, such as Solaris Volume Manager.

Before creating your first file system, you should familiarize yourself with file system layout possibilities. For information about volume management, file system layout, and other aspects of file system design, see the Sun QFS File System Configuration and Administration Guide.

> Note -
> If you are using a shared file system configuration that contains the Solaris 10 OS on both x64 platforms and SPARC platforms, Extensible Firmware Interface (EFI) labels are required on all shared disks. See Configuring EFI Labels for Shared x64 and SPARC Volumes for information about relabeling disks.

How to Estimate Disk Cache Requirements

Use the following guidelines to estimate the disk cache needed for SAM-QFS software (file systems plus the storage and archive manager):

- Disk cache = largest file (in bytes) + amount of space needed for working files
- Metadata cache
  Use the following information to estimate the metadata cache requirements. The metadata cache must have enough space to contain the following data:
  - Two copies of the superblock (16 Kbytes each)
  - Reservation maps for metadata space plus data space((metadata + file data)/disk allocation unit (DAU)/32,000) * 4 Kbytes
  - Inode space(number of files + number of directories) * 512 bytes
  - Indirect blocks – a minimum of 16 Kbytes each
  - Directory data space(number of directories * 16 Kbytes)

Run the `format`(1M) command to verify that you have sufficient disk cache space.

The `format`(1M) command shows how the disks are partitioned and the size of each partition.

Example 1 - Using the format(1M) Command on Fibre-Channel-Attached Disks

This example shows six disks attached to a server. Two internal disks are connected by means of controller 0 on targets 10 and 11 (`c0t10d0` and `c0t11d0`). The other disks are external.

For the sake of clarity, the `format`(1M) command output in this example has been edited.

Example – ***`format`***(1M) Command for Fibre-Channel-Attached Disks

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
       0. c0t10d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /sbus@3,0/SUNW,fas@3,8800000/sd@a,0
       1. c0t11d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /sbus@3,0/SUNW,fas@3,8800000/sd@b,0
       2. c9t60020F2000003A4C3ED20F150000DB7Ad0 <SUN-T300-0118 cyl 34530 alt 2 hd 48 sec 128>
          /scsi_vhci/ssd@g60020f2000003a4c3ed20f150000db7a
       3. c9t60020F2000003A4C3ED215D60001CF52d0 <SUN-T300-0118 cyl 34530 alt 2 hd 48 sec 128>
          /scsi_vhci/ssd@g60020f2000003a4c3ed215d60001cf52
       4. c9t60020F2000003A4C3ED21628000EE5A6d0 <SUN-T300-0118 cyl 34530 alt 2 hd 48 sec 128>
          /scsi_vhci/ssd@g60020f2000003a4c3ed21628000ee5a6
       5. c9t60020F2000003A4C3ED216500009D48Ad0 <SUN-T300-0118 cyl 34530 alt 2 hd 48 sec 128>
          /scsi_vhci/ssd@g60020f2000003a4c3ed216500009d48a
Specify disk (enter its number):^d
#
# format /dev/rdsk/c9t60020F2000003A4C3ED216500009D48Ad0s2
# *format f*
partition> p


Part      Tag    Flag     Cylinders         Size            Blocks
  0 unassigned    wm       0 -  4778       14.00GB    (4779/0/0)   29362176
  1 unassigned    wm    4779 -  9557       14.00GB    (4779/0/0)   29362176
  2     backup    wu       0 - 34529      101.16GB    (34530/0/0) 212152320
  3 unassigned    wm    9558 - 14336       14.00GB    (4779/0/0)   29362176
  4 unassigned    wm   14337 - 19115       14.00GB    (4779/0/0)   29362176
  5 unassigned    wm   19116 - 23894       14.00GB    (4779/0/0)   29362176
  6 unassigned    wm   23895 - 28673       14.00GB    (4779/0/0)   29362176
  7 unassigned    wm   28674 - 33452       14.00GB    (4779/0/0)   29362176

partition> ^D
#
```

Example 2 - Using the format(1M) Command on SCSI-Attached Disks

The following example shows four disks attached to a server. Two internal disks are connected by means of controller `0` on targets `0` (`c0t0d0`) and `1` (`c0t1d0`). Two external disks are connected by means of controller `3` on targets `0` (`c3t0d0`) and `2` (`c3t2d0`).

Example – ***format***(1M) Command for SCSI-Attached Disks

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
          /sbus@1f,0/SUNW,fas@e,8800000/sd@0,0
       1. c0t1d0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
          /sbus@1f,0/SUNW,fas@e,8800000/sd@1,0
       2. c3t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /sbus@1f,0/QLGC,isp@0,10000/sd@0,0
       3. c3t2d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /sbus@1f,0/QLGC,isp@0,10000/sd@2,0
Specify disk (enter its number): *1*
selecting c0t1d0
[disk formatted]
Warning: Current Disk has mounted partitions.

FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        repair     - repair a defective sector
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        inquiry    - show vendor, product and revision
        volname    - set 8-character volume name
        <cmd>      - execute <cmd>, then return
        quit
format> par

PARTITION MENU:
        0      - change "0" partition
        1      - change "1" partition
        2      - change "2" partition
        3      - change "3" partition
        4      - change "4" partition
        5      - change "5" partition
        6      - change "6" partition
        7      - change "7" partition
        select - select a predefined table
        modify - modify a predefined partition table
        name   - name the current table
        print  - display the current table
        label  - write partition map and label to the disk
        <cmd>  - execute <cmd>, then return
        quit
partition> pri
Current partition table (original):
Total disk cylinders available: 2733 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0        var    wm       0 - 2732        1.98GB    (2733/0/0) 4154160
  1 unassigned    wm       0               0         (0/0/0)          0
  2     backup    wm       0 - 2732        1.98GB    (2733/0/0) 4154160
  3 unassigned    wm       0               0         (0/0/0)          0
  4 unassigned    wm       0               0         (0/0/0)          0
  5 unassigned    wm       0               0         (0/0/0)          0
  6 unassigned    wm       0               0         (0/0/0)          0
  7 unassigned    wm       0               0         (0/0/0)          0

partition> q
```

Verifying Disk Space

The software requires a disk cache consisting of redundant arrays of inexpensive disks (RAID) devices, JBOD ("just a bunch of disks") devices, or both. It also requires a certain amount of disk space in the / (root), /opt, and /var directories. The actual amount needed varies depending on the packages you install.

The following table shows the minimum amount of disk space required in these various directories.

Table – Minimum Disk Space Requirements

Directory	SAM (Archiving and File System)	QFS (File System Only)	SAM-QFS Manager
/ (root) directory	2 Mbytes	2 Mbytes	25 Mbytes
/opt directory	21 Mbytes	8 Mbytes	5 Mbytes
/var directory	4 Gbytes	1 Mbyte	2 Mbytes
/usr directory	2 Mbytes	2 Mbytes	7 Mbytes
/tmp directory	0 Mbytes	0 Mbytes	200 Mbytes

> Note -
> The space requirements for the /var directory take into account the fact that the archiver data directory, the archiver queue files, and the log files are written to the /var directory.

How to Verify Disk Space

The following procedure shows how to verify whether there is enough disk space on your system to accommodate the SUNWsamfsu and SUNWsamfsr software packages.

1. Run the following command to verify that at least 2 Mbytes are in the avail column for the / directory.

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

2. Run the following command to verify that at least 21 Mbytes are in the avail column for the /opt directory.

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

3. Verify that at least 6 Mbytes are available in the /var directory.
   A quantity of 30 Mbytes or more is recommended to allow for the growth of log files and other system files.
4. If each directory does not have enough room for the software, repartition the disk to make more space available to each file system.
   For information about how to repartition a disk, see your Sun Solaris system administration documentation.

Preparing Hardware for Archiving

Verifying Archive Media

If you plan to perform disk archiving (to archive to disk space in another file system), verify the following:

- The host system to which the disks are attached has at least one file system created that is compatible with the SAM-QFS software.
- The disk has enough space available to accommodate the archive copies.

If you plan to archive to removable media devices, your environment must include the following:

- At least one removable media device for archiving files. This device can be a single tape or optical drive, or it can be multiple devices, such as the drives within an automated library.
- Tape or magneto-optical cartridges to which archive files can be written. For most SCSI-attached and FC-attached libraries, the SAM-QFS software supports only one media type. If you have a tape library that can be partitioned logically into two or more libraries,

you can have one media type in one logical library and a different media type in another. The SAM-QFS software records the cartridges used for each library in a library catalog. You cannot mix the tape media types in a library catalog, so plan to use only one media type per library or logical library.

The SAM-QFS environment supports a wide variety of removable media devices. You can obtain a list of currently supported drives and libraries from your Sun Microsystems sales or support staff.

To make sure that your devices are attached and enumerated in an easily retrieved list, perform one or both of the following procedures:

- If your removable media devices are not attached to your server, perform the procedure in Verifying Archive Media.
- Enumerate your devices using the instructions in Creating a List of Devices. You will use this list again in Installing the Software Packages.

How to Attach Removable Media Devices

The following steps are general guidelines for attaching removable media hardware to a server. For explicit instructions on how to connect these peripherals to a server, refer to the hardware installation guide supplied by the vendor with the automated library and drives.

1. Ensure that you are on a console connection to the server.
2. Power off the server.
3. Ensure that the removable media devices and the disks to be used for the Sun QFS file system are connected and properly addressed.
4. If you have libraries attached to the host system through a SCSI interface, ensure that the SCSI target IDs are unique for each SCSI initiator (host adapter).
   Avoid setting SCSI target IDs for peripherals to IDs that are already in use. In addition, if you are using a SCSI host adapter with a previously attached disk drive, any additional peripheral connected to this bus must have a different ID. Typically, the initiator uses ID 7, and the internal disk drive uses ID 3 for SPARC systems and ID 0 for UltraSPARC systems.
5. Power on the peripherals according to the manufacturer's recommended sequence.
   Typically, you power on the outermost peripherals first, working toward more central components in sequence.
6. Disable autobooting.
   At the >ok prompt, type the following command to disable autobooting:

   ```
   >ok setenv auto-boot? false
   ```

7. Type `reset` at the next prompt:

   ```
   >ok reset
   ```

8. Do one of the following:
   - If you have libraries attached to the host system through a SCSI interface, use the `probe-scsi-all` command to conduct an inventory of target IDs and logical unit numbers (LUNs) for each device connected to the host system. Save the output. You will use the information in this output for the next procedure, Creating a List of Devices.
     For example:

     ```
     {0} ok probe-scsi-all
     /pci@6,400/scsi@2,1
     Target 0
       Unit 0   Removable Device type 8     STK 9730        1700
     Target 1
       Unit 0   Removable Tape   type 7     QUANTUM DLT7000 2565
     Target 2
       Unit 0   Removable Tape   type 7     QUANTUM DLT7000 2565
     /pci@1f,4000/scsi@3
     Target 0
       Unit 0   Disk     SEAGATE ST318404LSUN18G 4207
     Target 6
       Unit 0   Removable Read Only device   TOSHIBA XM6201TASUN32XCD1103
     ```

   - If libraries or tape drives are attached to the host system through an FC interface, conduct an inventory of target IDs and LUNs for each device connected to the host system. Save the output. You will use the information in this output for the next procedure, Creating a List of Devices.
     For example:

```
{0} ok show-devs
/SUNW,ffb@1e,0
/SUNW,UltraSPARC-II@2,0
/SUNW,UltraSPARC-II@0,0
/counter-timer@1f,1c00
/pci@1f,2000
/pci@1f,4000
/virtual-memory
/memory@0,a0000000
/aliases
/options
/openprom
/chosen
/packages
/pci@1f,2000/SUNW,qlc@1
/pci@1f,2000/SUNW,qlc@1/fp@0,0
/pci@1f,2000/SUNW,qlc@1/fp@0,0/disk
/pci@1f,4000/SUNW,ifp@2
/pci@1f,4000/scsi@3,1
/pci@1f,4000/scsi@3
/pci@1f,4000/network@1,1
/pci@1f,4000/ebus@1
/pci@1f,4000/SUNW,ifp@2/ses
{0} ok select /pci@1f,2000/SUNW,qlc@1
{0} ok show-children
LiD HA LUN  --- Port WWN ---   ----- Disk description -----
 2  7e   0  500104f00041182b  STK     L700            0236
7c  7e   0  500104f00043abfc  STK     9840            1.28
7d  7e   0  500104f00045eeaf  STK     9840            1.28
6f  7e   0  500104f000416304  IBM     ULT3580-TD1     16E0
6e  7e   0  500104f000416303  IBM     ULT3580-TD1     16E0
```

If the server does not acknowledge all the known devices (disk drives, tape or optical drives, the automated library, and so on), check the cabling. Do not proceed until all devices appear when probed.

9. Reenable autobooting, and then boot the system:

```
>ok setenv auto-boot? true
>ok *boot*
```

10. Review system files.
Review the following files:
    - `/var/adm/messages` to ensure that all devices were recognized
    - `/dev/rmt` for expected tape devices
    - `/dev/dsk` and `/dev/rdsk` for expected disks
    Due to special driver requirements, no device information appears in `/var/adm/messages` for magneto-optical devices or libraries until after you install the SAM-QFS software packages.
11. Disable autocleaning and autoloading.
If your automated library supports autocleaning or autoloading, disable those features when using that library with the Sun Storage Archive Manager software. Consult the documentation from your library's manufacturer for information about disabling autocleaning and autoloading.

> Note -
> The only times you can use autoloading are during the initial loading of cartridges and when the SAM-QFS software is not running. Remember to disable autoloading when the SAM-QFS system is running.

Creating a List of Devices

The devices that you intend to use must be attached and recognized by the server upon which you intend to install the SAM-QFS software. To configure the SAM-QFS software, you need to know the following about your devices:

- The device type, manufacturer, and model number.
- The mechanism by which the device is attached to the server. You can attach devices in one of the following ways:

- Drives can use either a SCSI attachment or an FC attachment. Each drive accepts either tape cartridges or magneto-optical cartridges.
  For SCSI-attached drives, you need to know each drive's SCSI target ID and logical unit number (LUN).
  For FC-attached drives, you need to know each drive's LUN and node World Wide Name (WWN).
- Automated libraries can use a SCSI attachment, an FC attachment, or a network attachment.
  Libraries that use SCSI or FC attachments are called direct attached libraries. For SCSI-attached libraries, you need to know each library's SCSI target ID and LUN. For FC-attached libraries, you need to know each library's LUN and node WWN.
  Libraries that use a network attachment are called network attached libraries. You cannot configure network attached libraries in the existing system configuration files; instead, you must create a parameters file for each network attached library. This is explained later in the installation process.

How to Create a List of Devices

Fill in the following table to include the name, manufacturer, model, and connection types for each device that you want to include in your SAM-QFS environment. Retain this list for use again later in the configuration procedure.

Device Name, Manufacturer, and Model	Target ID	LUN	Node WWN
SCSI-attached tape drives			
			Not applicable
			Not applicable
			Not applicable
FC-attached tape drives			
	Not applicable		
	Not applicable		
	Not applicable		
SCSI-attached magneto-optical drives			
			Not applicable
			Not applicable
			Not applicable
FC-attached magneto-optical drives			
	Not applicable		
	Not applicable		
	Not applicable		
SCSI-attached automated libraries			
			Not applicable
			Not applicable
			Not applicable
FC-attached automated libraries			
	Not applicable		
	Not applicable		
	Not applicable		

Obtaining the Release Files

Make sure that you have a copy of the release software. You can obtain the Sun SAM-QFS software from the Sun Download Center or on a

CD-ROM. Contact your authorized service provider (ASP) or your Sun sales representative if you have questions on obtaining the software.

After the release, upgrade patches are available from Sun Solve.

> Caution -
> If you have not read the Release Notes, do so before you continue.

How to Obtain the Software From the Sun Download Center

1. Go to the downloads page.
2. Select the SAM-QFS or Sun QFS software package you want to receive.
3. Follow the instructions on the web site for downloading the software.

Software Licensing

You must agree to all binary and right-to-use (RTU) software license agreements before you install either the Sun QFS or the Sun SAM software.

Setting Up the Network Management Station

Perform this procedure if you want to monitor your configuration through Simple Network Management Protocol (SNMP) software.

You can configure the Sun SAM-QFS software to notify you when potential problems occur in its environment. The SNMP software manages information exchange between network devices such as servers, automated libraries, and drives. When the Sun SAM-QFS software detects potential problems in its environment, it sends information to a management station, which enables you to monitor the system remotely.

The management stations you can use include the following:

- Sun Storage Automated Diagnostic Environment (StorADE)
- Sun Management Center (Sun MC)
- Sun Remote Server (SRS)
- Sun Remote Services Net Connect

If you want to enable SNMP traps, make sure that the management station software is installed and operating correctly before installing the Sun SAM-QFS software. Refer to the documentation that came with your management station software.

The types of problems, or events, that the SAM-QFS software can detect are defined in the SAM-QFS Management Information Base (MIB). The events include errors in configuration, `tapealert`(1M) events, and other atypical system activity. For complete information about the MIB, see `/var/snmp/mib/SUN-SAM-MIB.mib` after the packages are installed.

The SAM-QFS software supports the TRAP SNMP (V2c) protocol. The software does not support `GET-REQUEST`, `GETNEXT-REQUEST`, and `SET_REQUEST`.

Next Steps

- Release Package Contents
- Installing the Software Packages

Installing the Software Packages

Installing the Software Packages

The Sun QFS and Sun Storage Archive Manager (SAM-QFS) software uses the Sun Solaris packaging utilities for adding and deleting software. The `pkgadd`(1M) utility prompts you to confirm various actions necessary to install the packages.

Table – Release Package Names

Installed Package	Description

SUNWqfsr SUNWqfsu	Sun QFS (file system only) software packages
SUNWsamfsr SUNWsamfsu	SAM-QFS(archiving and file system) software packages

How to Add the Packages

1. Become superuser.
2. Use the cd command to go to the directory where the software package release files reside.
   You obtained the release files as described in Obtaining the Release Files. Changing to the appropriate directory differs, depending on your release media, as follows:
   - If you downloaded the release files, go to the directory to which you downloaded the files.
   - If you obtained the release files from a CD-ROM, go to the directory on the CD-ROM that corresponds to your operating system version.
3. Use the pkgadd(1M) command to add the appropriate packages, based on the features that you need to support:
   - For archiving to a local or shared file system, install the SUNWsamfsr and SUNWsamfsu packages.
   - To support just a local or shared file system (no archiving), install the SUNWqfsr and SUNWqfsu packages.
     For example:

     ```
     # pkgadd -d . SUNWsamfsr SUNWsamfsu
     ```
4. When prompted to define an administrator group, select yes or y to accept the default (no administrator group), or select no or n if you want to define an administrator group.
   You can reset permissions on certain commands later by using the set_admin(1M) command. For more information about this command, see Adding the Administrator Group or the set_admin(1M) man page.
5. Examine the SAM-QFS installation log file /tmp/SAM_install.log.
   This file should show that the pkgadd(1M) command added the SUNWsamfsr and SUNWsamfsu software packages. Verify that the SAM-QFS samst driver is also installed. If all files installed properly, the following message appears:

   ```
   Restarting the sysevent daemon
   ```

How to Set Up PATH and MANPATH Variables

To enable access to the commands and man pages for the Sun QFS and SAM-QFS commands, you must modify your PATH and MANPATH environment variables.

1. For users who will need to access the user commands, such as sls(1), add /opt/SUNWsamfs/bin to the users' PATH variables.
2. Edit your system setup files to include the correct paths to commands and man pages.
   - In the Bourne or Korn shells, change the PATH and MANPATH variables in the .profile file and export the variables.
     The following example shows how your .profile file might look after editing.

     ```
     PATH=$PATH:/opt/SUNWsamfs/bin:/opt/SUNWsamfs/sbin
     MANPATH=$MANPATH:/opt/SUNWsamfs/man
     export PATH MANPATH
     ```
   - In the C shell, edit the .login and .cshrc files.
     The path statement in your .cshrc file might look like the following example:

     ```
     set path = ($path /opt/SUNWsamfs/bin /opt/SUNWsamfs/sbin)
     ```

     The MANPATH statement in your .login file might look like the following example:

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

Configuring the File System Environment

Contents

Configuring the File System Environment

Each SAM-QFS software environment is unique. The system requirements and hardware differ from site to site. SAM-QFS environments support a wide variety of tape and optical devices, automated libraries, and disk drives. The system administrator at your site must set up the specific configuration for your environment.

The master configuration file, `/etc/opt/SUNWsamfs/mcf`, defines the equipment topology managed by the SAM-QFS software. This file specifies the devices, automated libraries, and file systems included in the environment. You assign each piece of equipment a unique Equipment Identifier in the `mcf` file.

You can edit the `mcf` file in either of two ways:

- By using the SAM-QFS Manager interface to configure archiving and file system devices. When you create a file system using SAM-QFS Manager, it creates an `mcf` file in `/etc/opt/SUNWsamfs/mcf` that contains a line for each device and family set of the file system.

- By directly editing the `mcf` file using a text editor.

The `mcf` file has two kinds of entries:

- File system device entries for disk devices. In the `mcf` file, you organize them into one or more file systems.
- Removable media device entries that you can organize into family sets. The `mcf` file contains information that enables you to identify the drives to be used and associate them with the automated libraries to which they are attached.

For detailed information about `mcf` file structures and contents, see About the Master Configuration File.

Example `mcf` files are installed in `/opt/SUNWsamfs/examples`. Several example `mcf` file configurations are also provided in Examples of `mcf` Files.

The following sections provide examples and describe activities related to creating and maintaining the `mcf` file.

How to Create the Master Configuration File (mcf) Manually

- Use vi(1) or another editor to create the `/etc/opt/SUNWsamfs/mcf` file.
  For detailed information about the contents of the `mcf` file, see About the Master Configuration File.

  > You can copy an example `mcf` file from `/opt/SUNWsamfs/examples` or from the examples in Examples of `mcf` Files.

  When you create the `mcf` file, follow these guidelines:

  - Delimit the fields in each line with spaces or tabs.
  - Begin each comment line entered into this file with a pound sign (#).
  - Use a dash (-) to indicate optional fields that are omitted.
    The following example shows the `mcf` file fields.

```
#
# Sun  Storage Archive Manager file system configuration
#
# Equipment       Equip  Equip Fam   Dev    Additional
# Identifier      Ord    Type  Set   State  Parameters
# ----------      -----  ----- ----  -----  ----------
```

The mcf file can contain both comment lines and lines that pertain to a device. The types of lines that can pertain to a device are as follows:

- Family set parent identifiers and family set devices
- Family set member devices
- Stand-alone devices

Identifying Peripherals Using the /var/adm/messages File

When your system boots, a series of messages is written to /var/adm/messages. These messages identify the Sun Solaris hardware path to each of the peripherals on your system. You can use this information to create the mcf file. To display information from the latest system reboot, search backward from the end of the file.

As the following example shows, each SCSI peripheral has three lines. The sixth field, samst2, indicates that these lines are associated with each other.

Example – SCSI Peripheral Lines in the ***/var/adm/messages*** File

```
# tail -200 /var/adm/messages | more
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@5,8800000/samst@2,0
```

- The first line displays the vendor and product information that the SCSI peripheral reported to the Sun Solaris kernel.
- The second line displays the SCSI bus, SCSI target ID, and LUN of the peripheral.
- The third line displays the peripheral's hardware path. This path is reflected in the /devices directory. Symbolic links (symlinks) to the /devices directory are set up in the /dev/st, /dev/samst, and /dev/rmt directories. Note that the third line wraps to the next line.

Matching the symbolic link to the correct peripheral is the key to configuring a Sun Storage Archive Manager environment. Use the ls(1) command with the -l option in both the /dev/st, /dev/samst and /dev/rmt directories to display the path name of the peripheral.

You might also want to set up the "device down" notification script at this point. The dev_down.sh(1M) man page contains information about setting up this script, which sends email to root when a device is marked down or off. For more information, see the dev_down.sh(1M) man page.

How to Verify the mcf File

- If you created your mcf file manually, use the sam-fsd(1M) command to verify the file.
  If you created your mcf file using SAM-QFS Manager, you do not need to verify its syntax.
  If the mcf file is free of syntax errors, the sam-fsd(1M) output includes information about the file systems, archiving, and other system information. If the mcf file contains syntax or other errors, however, the output is similar to the following example.

```
# sam-fsd
13: /dev/dsk/c1t1d0s0   10      md      samfs1  on      /dev/rdsk/c1t1d0s0
*** Error in line 13: Equipment name '/dev/dsk/c1t1d0s0' already in use by eq 10
72: /dev/rmt/3cbn       45      ug      l1000   on
*** Error in line 72: Equipment name '/dev/rmt/3cbn' already in use by eq 44
2 errors in '/etc/opt/SUNWsamfs/mcf'
sam-fsd: Read mcf /etc/opt/SUNWsamfs/mcf failed.
```

If the `mcf` file has errors, return to Setting Up the Environment Configuration and refer to the `mcf(4)` man page for information about creating this file. You can also refer to Examples of `mcf` Files.

How to Create an mcf File Using SAM-QFS Manager

Before You Begin

When you configure QFS file systems using the SAM-QFS Manager software, it creates or edits the appropriate configuration files, including the `mcf` file, on that server. You can use either SAM-QFS Manager or the CLI to further edit these files later.

> Note -
> If you want to use SAM-QFS Manager to configure your archiving environment and you want to include network attached libraries (excluding STK Libraries) in this configuration, you must create your parameters file before you create the `mcf` file. For information about creating a parameters file, see Creating Parameters Files for Network Attached Automated Libraries.
> You can add a StorageTek ACSLS network library in the SAM-QFS Manager without creating the parameters file. The application automatically generates the parameters file for you when you add the library in the Library Summary Page.

Steps

1. From your web browser, log in to the SAM-QFS Manager as an administrator user.
2. Expand the Getting Started section and choose First Time Configuration.
3. In section 2, click Create a File System.
   The New File System wizard is displayed.
4. Follow the steps for creating a new file system.
   When you have completed this process, the `mcf` file is created. For more information, see the SAM-QFS Manager online help.

Initializing the Environment

Initializing the Environment

This section tells you how to initialize the environment and the file system, and how to mount the file system.

How to Initialize the Environment

1. Use the `samd`(1M) `config` command to initialize the archiving and file system environment.
   For example:

```
# samd config
```

How to Initialize the File System

This procedure describes how to use the `sammkfs`(1M) command and the Family Set names that you have defined to initialize a file system.

> Note -
> The `sammkfs`(1M) command sets one tuning parameter, the disk allocation unit (DAU). You cannot reset this parameter without reinitializing the file system. For information about how the DAU affects tuning, see File Allocation Methods and the `sammkfs`(1M) man page.

1. Use the `sammkfs`(1M) command to initialize a file system for each Family Set name defined in the `mcf` file.

> Caution -
> Running the `sammkfs`(1M) command creates a new file system. It removes all references to the data currently contained in the partitions associated with the file system in the `/etc/opt/SUNWsamfs/mcf` file.

The following example shows the command to initialize a file system with the Family Set name of samfs1.

```
# sammkfs samfs1
sammkfs: Configuring file system
Building "samfs1" will destroy the contents of devices:
    /dev/dsk/c2t0d0s3
    /dev/dsk/c2t0d0s7
Do you wish to continue? [y/N] *y*
total data kilobytes       = 16777728
total data kilobytes free  = 16777152
#
```

The actual numbers returned vary from file system to file system.

Mounting the File System

The mount(1M) command mounts a file system and reads the /etc/vfstab and samfs.cmd configuration files. For information, see the mount_samfs(1M) man page.

How to Mount the File System Using SAM-QFS Manager

1. On the Managed Hosts page, click the name of the server on which the file system is located.
   The File Systems Summary page is displayed.
2. Select the radio button next to the file system that you want to mount.
3. From the Operations menu, choose Mount.

How to Mount the File System From the Command Line

1. Use the mount(1M) command to mount the file system.
   Specify the file system mount point as the argument. For example:

```
# mount /samfs1
```

> Tip -
> If the file system has not been added to the /etc/vfstab file, use the following form of the mount command:
>
> ```
> #mount -F samfs <fs_name> </mount_point>
> ```

2. Use the mount(1M) command with no arguments to verify the mount.
   This step confirms that the file system is mounted and shows how to set permissions. The following example shows the output from a mount(1M) command issued to verify whether example file system samfs1 is mounted.

```
# mount
_<<< information deleted >>>_
/samfs1 on samfs1 read/write/setuid/intr/largefiles/onerror=panic/dev=8001e3 on Thu Feb  5
11:01:23 2004
_<<< information deleted >>>_
```

3. (Optional) Issue the chmod(1) and chown(1) commands to change the permissions and ownership of the file system's root directory.
   Typically, this step is performed if this is the first time that the file system has been mounted. For example:

```
# chmod 755 /samfs1
# chown root:other /samfs1
```

Setting Up Mount Parameters

Setting Up Mount Parameters

Use the procedures in this section to specify mount parameters for the file system.

You can specify mount parameters in the following ways:

- Using the `mount`(1M) command. Mount options specified here override those specified in the `/etc/vfstab` file and in the `samfs.cmd` file.
- In the `/etc/vfstab` file. Mount options specified here override those specified in the `samfs.cmd` file.
- In the `samfs.cmd` file.

For a list of available mount options, see the `mount_samfs`(1M) man page.

Updating the /etc/vfstab File and Creating the Mount Point

This section describes how to edit the `/etc/vfstab` file.

The following table shows the values you can provide in the fields in the `/etc/vfstab` file.

Table – ***/etc/vfstab*** File Fields

Field	Field Title and Content
1	Device to Mount. The name of the file system to be mounted. This value must be the same as the file system's Family Set name specified in the `mcf` file.
2	Device to `fsck`(1M). Must be a dash (`-`) character, which indicates that there are no options. This character prevents the Solaris system from performing an `fsck`(1M) process on the file system. For more information about this process, see the `fsck`(1M) or `samfsck`(1M) man page.
3	Mount Point, for example, `/samfs1`.
4	File System Type. Must be `samfs`.
5	`fsck`(1M) Pass. Must be a dash (`-`) character, which indicates that there are no options.
6	Mount at Boot. Either yes or no. • Specifying `yes` in this field indicates that the Sun Storage Archive Manager file system is to be mounted automatically at boot time. • Specifying no in this field indicates that you do not want to mount the file system automatically. For information about the format of these entries, see the `mount_samfs`(1M) man page.
7	Mount Parameters. A list of comma-separated parameters (with no spaces) that are used in mounting the file system. You can specify mount options on the `mount`(1M) command, in the `/etc/vfstab` file, or in a `samfs.cmd` file. Mount options specified on the `mount`(1M) command override those specified in the `/etc/vfstab` file and in the `samfs.cmd` file. Mount options specified in the `/etc/vfstab` file override those in the `samfs.cmd` file. For a list of available mount options, see the `mount_samfs`(1M) man page.

When you create a file system using SAM-QFS Manager, a default `/etc/vfstab` file is created. However, mount options specified in SAM-QFS Manager are written to the `samfs.cmd` file rather than to the `/etc/vfstab` file. For more information, see Creating and Editing the `samfs.cmd` File.

To edit the mount options in the `/etc/vfstab` file, use the following procedure, How to Update the /etc/vfstab File and Create the Mount Point Using a Text Editor.

How to Update the /etc/vfstab File and Create the Mount Point Using a Text Editor

The example in this task assumes that `/samfs1` is the mount point of the `samfs1` file system.

1. In the `/etc/vfstab` file, create an entry for each file system.
   The following example shows header fields and entries for a local file system.

```
#DEVICE    DEVICE   MOUNT   FS    FSCK  MOUNT    MOUNT
#TO MOUNT  TO FSCK  POINT   TYPE  PASS  AT BOOT  PARAMETERS
#
samfs1     -        /samfs1 samfs  -    yes      high=80,low=60
```

2. Use the `mkdir`(1M) command to create the mount point.
   For example:

```
# mkdir /samfs1
```

Creating and Editing the samfs.cmd File

You can create the `/etc/opt/SUNWsamfs/samfs.cmd` file as the place from which the system reads mount parameters. If you are configuring multiple file systems with multiple mount parameters, consider creating this file.

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

How to Create and Edit the samfs.cmd File Using SAM-QFS Manager

If you specify non-default mount options when creating a file system in SAM-QFS Manager, the `samfs.cmd` file is created or updated with those mount options.

Follow these steps to edit a file system's mount options:

1. On the Managed Hosts page, click the name of the server on which the file system is located.
   The File Systems Summary page is displayed.
2. Select the radio button next to the file system whose mount options you want to edit.
3. From the Operations menu, choose Edit Mount Options.
   The Edit Mount Options page is displayed.
4. Make your edits in the fields.
   For more information about the fields on the Edit Mount Options page, see the File System Manager online help.
5. Click Save.
   The new mount options are written to the `samfs.cmd` file.

How to Create and Edit the samfs.cmd File Using a Text Editor

- Use `vi`(1) or another editor to create the `samfs.cmd` file.
  Create lines in the `samfs.cmd` file to control mounting, performance features, or other aspects of file system management. For more information about the `samfs.cmd` file, see The samfs.cmd File or the `samfs.cmd`(4) man page.

Installing Sun QFS

Contents

Complete (Printable) Sun QFS 5.1 Installation Guide

Installing Sun QFS

Complete the tasks below if this is the initial installation of the Sun QFS (file system only) software packages at your site.

- To upgrade Sun QFS software on an existing server, see Upgrading QFS and SAM-QFS.
- To install the software in a Sun Cluster environment, you must also follow the additional instructions in Using SAM-QFS With Solaris Cluster.

> Note
> You must be logged in as superuser to complete the installation tasks.
> Installing QFS involves many of the same steps as installing SAM-QFS, except that you do not configure the archiving storage devices.

Before You Begin

- If you are not already familiar with the QFS file system, read the File System Overview.
- Before you follow the detailed installation steps below, check the hardware and software requirements as explained in Preparing for Installation.

Installation Overview Task Map

Depending on the the features that you need to support, you must complete several of the following procedures.

Step	Task	Description
1	Add the software packages.	Install the appropriate packages for your needs. Also see Release Package Contents.
2	Configure `path` and `manpath` variables.	Configure the environment variables for access to commands and man pages.
3	(Optional) Install and configure SAM-QFS Manager.	This task is needed only if you want to use a browser to configure file systems.
4	Configure the file system environment.	Define the master configuration file `mcf`.
5	Configure file system mount parameters.	Define the `/etc/vfstab` and `samfs.cmd` files.
6	Initialize the environment.	Initialize SAM-QFS and mount the file systems.

7	(Optional) Configure shared file systems.	If applicable to your environment, complete the configuration tasks specific to a shared Sun QFS environment (see Configuring a Shared File System).
8	(Optional} Configure high-availability for your file systems.	If applicable to your environment, complete the configuration tasks specific to a Sun Cluster environment (see Using SAM-QFS With Solaris Cluster).

Installing the Software Packages

The Sun QFS and Sun Storage Archive Manager (SAM-QFS) software uses the Sun Solaris packaging utilities for adding and deleting software. The pkgadd(1M) utility prompts you to confirm various actions necessary to install the packages.

Table – Release Package Names

Installed Package	Description
SUNWqfsr SUNWqfsu	Sun QFS (file system only) software packages
SUNWsamfsr SUNWsamfsu	SAM-QFS(archiving and file system) software packages

How to Add the Packages

1. Become superuser.
2. Use the cd command to go to the directory where the software package release files reside.
   You obtained the release files as described in Obtaining the Release Files. Changing to the appropriate directory differs, depending on your release media, as follows:
   - If you downloaded the release files, go to the directory to which you downloaded the files.
   - If you obtained the release files from a CD-ROM, go to the directory on the CD-ROM that corresponds to your operating system version.
3. Use the pkgadd(1M) command to add the appropriate packages, based on the features that you need to support:
   - For archiving to a local or shared file system, install the SUNWsamfsr and SUNWsamfsu packages.
   - To support just a local or shared file system (no archiving), install the SUNWqfsr and SUNWqfsu packages.
     For example:

```
# pkgadd -d . SUNWsamfsr SUNWsamfsu
```

4. When prompted to define an administrator group, select yes or y to accept the default (no administrator group), or select no or n if you want to define an administrator group.
   You can reset permissions on certain commands later by using the set_admin(1M) command. For more information about this command, see Adding the Administrator Group or the set_admin(1M) man page.
5. Examine the SAM-QFS installation log file /tmp/SAM_install.log.
   This file should show that the pkgadd(1M) command added the SUNWsamfsr and SUNWsamfsu software packages. Verify that the SAM-QFS samst driver is also installed. If all files installed properly, the following message appears:

```
Restarting the sysevent daemon
```

How to Set Up PATH and MANPATH Variables

To enable access to the commands and man pages for the Sun QFS and SAM-QFS commands, you must modify your PATH and MANPATH environment variables.

1. For users who will need to access the user commands, such as sls(1), add /opt/SUNWsamfs/bin to the users' PATH variables.
2. Edit your system setup files to include the correct paths to commands and man pages.
   - In the Bourne or Korn shells, change the PATH and MANPATH variables in the .profile file and export the variables.
     The following example shows how your .profile file might look after editing.

```
PATH=$PATH:/opt/SUNWsamfs/bin:/opt/SUNWsamfs/sbin
MANPATH=$MANPATH:/opt/SUNWsamfs/man
export PATH MANPATH
```

- In the C shell, edit the `.login` and `.cshrc` files.
  The `path` statement in your `.cshrc` file might look like the following example:

```
set path = ($path /opt/SUNWsamfs/bin /opt/SUNWsamfs/sbin)
```

  The `MANPATH` statement in your `.login` file might look like the following example:

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

Configuring the File System Environment

Each SAM-QFS software environment is unique. The system requirements and hardware differ from site to site. SAM-QFS environments support a wide variety of tape and optical devices, automated libraries, and disk drives. The system administrator at your site must set up the specific configuration for your environment.

The master configuration file, `/etc/opt/SUNWsamfs/mcf`, defines the equipment topology managed by the SAM-QFS software. This file specifies the devices, automated libraries, and file systems included in the environment. You assign each piece of equipment a unique Equipment Identifier in the `mcf` file.

You can edit the `mcf` file in either of two ways:

- By using the SAM-QFS Manager interface to configure archiving and file system devices. When you create a file system using SAM-QFS Manager, it creates an `mcf` file in `/etc/opt/SUNWsamfs/mcf` that contains a line for each device and family set of the file system.
- By directly editing the `mcf` file using a text editor.

The `mcf` file has two kinds of entries:

- File system device entries for disk devices. In the `mcf` file, you organize them into one or more file systems.
- Removable media device entries that you can organize into family sets. The `mcf` file contains information that enables you to identify the drives to be used and associate them with the automated libraries to which they are attached.

For detailed information about `mcf` file structures and contents, see About the Master Configuration File.

Example `mcf` files are installed in `/opt/SUNWsamfs/examples`. Several example `mcf` file configurations are also provided in Examples of `mcf` Files.

The following sections provide examples and describe activities related to creating and maintaining the `mcf` file.

How to Create the Master Configuration File (mcf) Manually

- Use vi(1) or another editor to create the `/etc/opt/SUNWsamfs/mcf` file.
  For detailed information about the contents of the `mcf` file, see About the Master Configuration File.

  > You can copy an example `mcf` file from `/opt/SUNWsamfs/examples` or from the examples in Examples of `mcf` Files.

  When you create the `mcf` file, follow these guidelines:

  - Delimit the fields in each line with spaces or tabs.

- Begin each comment line entered into this file with a pound sign (#).
- Use a dash (-) to indicate optional fields that are omitted.
  The following example shows the mcf file fields.

```
#
# Sun  Storage Archive Manager file system configuration
#
# Equipment       Equip  Equip Fam   Dev    Additional
# Identifier      Ord    Type  Set   State  Parameters
# ----------      -----  ----- ----  -----  ----------
```

The mcf file can contain both comment lines and lines that pertain to a device. The types of lines that can pertain to a device are as follows:

- Family set parent identifiers and family set devices
- Family set member devices
- Stand-alone devices

Identifying Peripherals Using the /var/adm/messages File

When your system boots, a series of messages is written to /var/adm/messages. These messages identify the Sun Solaris hardware path to each of the peripherals on your system. You can use this information to create the mcf file. To display information from the latest system reboot, search backward from the end of the file.

As the following example shows, each SCSI peripheral has three lines. The sixth field, samst2, indicates that these lines are associated with each other.

Example – SCSI Peripheral Lines in the **/var/adm/messages** File

```
# tail -200 /var/adm/messages | more
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@5,8800000/samst@2,0
```

- The first line displays the vendor and product information that the SCSI peripheral reported to the Sun Solaris kernel.
- The second line displays the SCSI bus, SCSI target ID, and LUN of the peripheral.
- The third line displays the peripheral's hardware path. This path is reflected in the /devices directory. Symbolic links (symlinks) to the /devices directory are set up in the /dev/st, /dev/samst, and /dev/rmt directories. Note that the third line wraps to the next line.

Matching the symbolic link to the correct peripheral is the key to configuring a Sun Storage Archive Manager environment. Use the ls(1) command with the -l option in both the /dev/st, /dev/samst and /dev/rmt directories to display the path name of the peripheral.

You might also want to set up the "device down" notification script at this point. The dev_down.sh(1M) man page contains information about setting up this script, which sends email to root when a device is marked down or off. For more information, see the dev_down.sh(1M) man page.

How to Verify the mcf File

- If you created your mcf file manually, use the sam-fsd(1M) command to verify the file.
  If you created your mcf file using SAM-QFS Manager, you do not need to verify its syntax.
  If the mcf file is free of syntax errors, the sam-fsd(1M) output includes information about the file systems, archiving, and other system information. If the mcf file contains syntax or other errors, however, the output is similar to the following example.

```
# sam-fsd
13: /dev/dsk/c1t1d0s0   10      md      samfs1  on      /dev/rdsk/c1t1d0s0
*** Error in line 13: Equipment name '/dev/dsk/c1t1d0s0' already in use by eq 10
72: /dev/rmt/3cbn       45      ug      l1000   on
*** Error in line 72: Equipment name '/dev/rmt/3cbn' already in use by eq 44
2 errors in '/etc/opt/SUNWsamfs/mcf'
sam-fsd: Read mcf /etc/opt/SUNWsamfs/mcf failed.
```

If the `mcf` file has errors, return to Setting Up the Environment Configuration and refer to the `mcf`(4) man page for information about creating this file. You can also refer to Examples of `mcf` Files.

How to Create an mcf File Using SAM-QFS Manager

Before You Begin

When you configure QFS file systems using the SAM-QFS Manager software, it creates or edits the appropriate configuration files, including the `mcf` file, on that server. You can use either SAM-QFS Manager or the CLI to further edit these files later.

> Note -
> If you want to use SAM-QFS Manager to configure your archiving environment and you want to include network attached libraries (excluding STK Libraries) in this configuration, you must create your parameters file before you create the `mcf` file. For information about creating a parameters file, see Creating Parameters Files for Network Attached Automated Libraries.
> You can add a StorageTek ACSLS network library in the SAM-QFS Manager without creating the parameters file. The application automatically generates the parameters file for you when you add the library in the Library Summary Page.

Steps

1. From your web browser, log in to the SAM-QFS Manager as an administrator user.
2. Expand the Getting Started section and choose First Time Configuration.
3. In section 2, click Create a File System.
   The New File System wizard is displayed.
4. Follow the steps for creating a new file system.
   When you have completed this process, the `mcf` file is created. For more information, see the SAM-QFS Manager online help.

Setting Up Mount Parameters

Use the procedures in this section to specify mount parameters for the file system.

You can specify mount parameters in the following ways:

- Using the `mount`(1M) command. Mount options specified here override those specified in the `/etc/vfstab` file and in the `samfs.cmd` file.
- In the `/etc/vfstab` file. Mount options specified here override those specified in the `samfs.cmd` file.
- In the `samfs.cmd` file.

For a list of available mount options, see the `mount_samfs`(1M) man page.

Updating the /etc/vfstab File and Creating the Mount Point

This section describes how to edit the `/etc/vfstab` file.

The following table shows the values you can provide in the fields in the `/etc/vfstab` file.

Table – **`/etc/vfstab`** File Fields

Field	Field Title and Content

1	Device to Mount. The name of the file system to be mounted. This value must be the same as the file system's Family Set name specified in the `mcf` file.
2	Device to `fsck`(1M). Must be a dash (`-`) character, which indicates that there are no options. This character prevents the Solaris system from performing an `fsck`(1M) process on the file system. For more information about this process, see the `fsck`(1M) or `samfsck`(1M) man page.
3	Mount Point, for example, `/samfs1`.
4	File System Type. Must be `samfs`.
5	`fsck`(1M) Pass. Must be a dash (`-`) character, which indicates that there are no options.
6	Mount at Boot. Either yes or no. • Specifying `yes` in this field indicates that the Sun Storage Archive Manager file system is to be mounted automatically at boot time. • Specifying no in this field indicates that you do not want to mount the file system automatically. For information about the format of these entries, see the `mount_samfs`(1M) man page.
7	Mount Parameters. A list of comma-separated parameters (with no spaces) that are used in mounting the file system. You can specify mount options on the `mount(1M)` command, in the `/etc/vfstab` file, or in a `samfs.cmd` file. Mount options specified on the `mount`(1M) command override those specified in the `/etc/vfstab` file and in the `samfs.cmd` file. Mount options specified in the `/etc/vfstab` file override those in the `samfs.cmd` file. For a list of available mount options, see the `mount_samfs`(1M) man page.

When you create a file system using SAM-QFS Manager, a default `/etc/vfstab` file is created. However, mount options specified in SAM-QFS Manager are written to the `samfs.cmd` file rather than to the `/etc/vfstab` file. For more information, see Creating and Editing the `samfs.cmd` File.

To edit the mount options in the `/etc/vfstab` file, use the following procedure, How to Update the /etc/vfstab File and Create the Mount Point Using a Text Editor.

How to Update the /etc/vfstab File and Create the Mount Point Using a Text Editor

The example in this task assumes that `/samfs1` is the mount point of the `samfs1` file system.

1. In the `/etc/vfstab` file, create an entry for each file system.
   The following example shows header fields and entries for a local file system.

```
#DEVICE    DEVICE    MOUNT   FS    FSCK  MOUNT    MOUNT
#TO MOUNT  TO FSCK   POINT   TYPE  PASS  AT BOOT  PARAMETERS
#
samfs1     -         /samfs1 samfs  -    yes      high=80,low=60
```

2. Use the `mkdir`(1M) command to create the mount point.
   For example:

```
# mkdir /samfs1
```

Creating and Editing the samfs.cmd File

You can create the `/etc/opt/SUNWsamfs/samfs.cmd` file as the place from which the system reads mount parameters. If you are configuring multiple file systems with multiple mount parameters, consider creating this file.

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

How to Create and Edit the samfs.cmd File Using SAM-QFS Manager

If you specify non-default mount options when creating a file system in SAM-QFS Manager, the `samfs.cmd` file is created or updated with those mount options.

Follow these steps to edit a file system's mount options:

1. On the Managed Hosts page, click the name of the server on which the file system is located.
   The File Systems Summary page is displayed.
2. Select the radio button next to the file system whose mount options you want to edit.
3. From the Operations menu, choose Edit Mount Options.
   The Edit Mount Options page is displayed.
4. Make your edits in the fields.
   For more information about the fields on the Edit Mount Options page, see the File System Manager online help.
5. Click Save.
   The new mount options are written to the `samfs.cmd` file.

How to Create and Edit the samfs.cmd File Using a Text Editor

- Use `vi`(1) or another editor to create the `samfs.cmd` file.
  Create lines in the `samfs.cmd` file to control mounting, performance features, or other aspects of file system management. For more information about the `samfs.cmd` file, see The samfs.cmd File or the `samfs.cmd`(4) man page.

Initializing the Environment

This section tells you how to initialize the environment and the file system, and how to mount the file system.

How to Initialize the Environment

1. Use the `samd`(1M) `config` command to initialize the archiving and file system environment.
   For example:

```
# samd config
```

How to Initialize the File System

This procedure describes how to use the `sammkfs`(1M) command and the Family Set names that you have defined to initialize a file system.

> Note -
> The `sammkfs`(1M) command sets one tuning parameter, the disk allocation unit (DAU). You cannot reset this parameter without reinitializing the file system. For information about how the DAU affects tuning, see File Allocation Methods and the `sammkfs`(1M) man page.

1. Use the `sammkfs`(1M) command to initialize a file system for each Family Set name defined in the `mcf` file.

   > Caution -
   > Running the `sammkfs`(1M) command creates a new file system. It removes all references to the data currently contained in the partitions associated with the file system in the `/etc/opt/SUNWsamfs/mcf` file.

   The following example shows the command to initialize a file system with the Family Set name of `samfs1`.

```
# sammkfs samfs1
sammkfs: Configuring file system
Building "samfs1" will destroy the contents of devices:
    /dev/dsk/c2t0d0s3
    /dev/dsk/c2t0d0s7
Do you wish to continue? [y/N] *y*
total data kilobytes       = 16777728
total data kilobytes free  = 16777152
#
```

The actual numbers returned vary from file system to file system.

Mounting the File System

The `mount`(1M) command mounts a file system and reads the `/etc/vfstab` and `samfs.cmd` configuration files. For information, see the `mount_samfs`(1M) man page.

How to Mount the File System Using SAM-QFS Manager

1. On the Managed Hosts page, click the name of the server on which the file system is located.
   The File Systems Summary page is displayed.
2. Select the radio button next to the file system that you want to mount.
3. From the Operations menu, choose Mount.

How to Mount the File System From the Command Line

1. Use the `mount`(1M) command to mount the file system.
   Specify the file system mount point as the argument. For example:

```
# mount /samfs1
```

> Tip -
> If the file system has not been added to the `/etc/vfstab` file, use the following form of the `mount` command:
>
> ```
> #mount -F samfs <fs_name> </mount_point>
> ```

2. Use the `mount`(1M) command with no arguments to verify the mount.
   This step confirms that the file system is mounted and shows how to set permissions. The following example shows the output from a `mount`(1M) command issued to verify whether example file system `samfs1` is mounted.

```
# mount
_<<< information deleted >>>_
/samfs1 on samfs1 read/write/setuid/intr/largefiles/onerror=panic/dev=8001e3 on Thu Feb  5
11:01:23 2004
_<<< information deleted >>>_
```

3. (Optional) Issue the `chmod`(1) and `chown`(1) commands to change the permissions and ownership of the file system's `root` directory.
   Typically, this step is performed if this is the first time that the file system has been mounted. For example:

```
# chmod 755 /samfs1
# chown root:other /samfs1
```

Related Topics

- Release Package Contents
- Upgrading QFS and SAM-QFS

Next Steps

- Installing SAM-QFS Manager
- Configuring the File System
- Configuring a Shared File System

Performing Additional SAM-QFS Configuration

Contents

Performing Additional SAM-QFS Configuration

This section outlines additional tasks that you might need to complete in order to finish the configuration of the Sun Storage Archive Manager (SAM-QFS) or Sun QFS environment. Some of these tasks are optional, depending on your specific environment.

Before You Begin

- Install the software as described in Installing and Configuring SAM-QFS or Installing Sun QFS, as appropriate.

Sharing the File System With NFS Client Systems

Perform this task if you want the file system to be shared with network file system (NFS) clients.

How to NFS Share the File System

This procedure uses the Solaris `share`(1M) command to make the file system available for mounting by remote systems. The `share`(1M) commands are typically placed in the `/etc/dfs/dfstab` file and are executed automatically by the Solaris OS when you enter `init`(1M) state 3.

Steps

1. Use `vi`(1) or another editor to add a `share`(1M) command to the `/etc/dfs/dfstab` file.
   For example:

```
share -F nfs -o rw=client1:client2 -d "SAM-FS" /samfs1
```

2. Use the ps(1) command to determine whether `nfs.server` is running.
   For example:

```
# ps -ef | grep nfsd
    root   694     1  0   Apr 29 ?        0:36 /usr/lib/nfs/nfsd -a 16
en17     29996 29940  0 08:27:09 pts/5    0:00 grep nfsd
# ps -ef | grep mountd
    root   406     1  0   Apr 29 ?       95:48 /usr/lib/autofs/automountd
    root   691     1  0   Apr 29 ?        2:00 /usr/lib/nfs/mountd
en17     29998 29940  0 08:27:28 pts/5    0:00 grep mountd
```

   In this sample output, the lines that contain `/usr/lib/nfs` indicate that the NFS server is mounted.

3. If `nfs.server` is not running, start it:

```
# /etc/init.d/nfs.server start
```

4. (Optional) Type the `share`(1M) command at a `root` shell prompt.
   Perform this step if you want to NFS-share the file system immediately.
   If no NFS-shared file systems exist when the Sun Solaris OS boots, the NFS server is not started. The following example shows the commands to use to enable NFS sharing. You must change to run level 3 after adding the first share entry to this file.

```
# init 3
# who -r
.       run-level 3  Dec 12 14:39     3    2  2
# share
-          /samfs1  -   "SAM-FS"
```

   Some NFS mount parameters can affect the performance of an NFS-mounted Sun Storage Archive Manager file system. You can set these parameters in the `/etc/vfstab` file as follows:

   - `timeo` = n. This value sets the NFS timeout to n tenths of a second. The default is eleven tenths of a second. For performance purposes, Sun Microsystems recommends using the default value. You can increase or decrease the value appropriately to your system.
   - `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.
     For more information about these parameters, see the `mount_nfs`(1M) man page.

How to Mount the File System on Clients

On the client systems, mount the server's file system at a convenient mount point.

> Note -
> There can be a significant delay in the file system's response to NFS client requests if a requested file resides on a cartridge that must be loaded into a DLT tape drive, if all tape drives are full, or if drives are slow. As a consequence, the system might generate an error instead of retrying the operation.
> To avoid this situation, it is recommended that you mount the file system with either the `hard` option enabled or with the `soft`, `retrans`, and `timeo` options enabled. If you use the `soft` option, also specify `retrans=120` (or greater) and `timeo=3000`.

Steps

1. On an NFS client system, use `vi`(1) or another editor to edit the `/etc/vfstab` file and add a line to mount the server's file system at a convenient mount point.
   The following example mounts `server:/samfs1` on the `/samfs1` mount point:

```
server:/samfs1   -   /samfs1   nfs -      yes   hard,intr,timeo=60
```

2. Save and close the /etc/vfstab file.
3. Enter the mount(1M) command.
   For example, the following mount(1M) command mounts the samfs1 file system:

```
client# mount /samfs1
```

   Alternatively, the automounter can do this, if you prefer. Follow your site procedures for adding server :/samfs1 to your automounter maps. For more information, see the automountd(1M) man page.

Editing the **defaults.conf** File

The /opt/SUNWsamfs/examples/defaults.conf file contains directives that control automated library actions in a Sun Storage Archive Manager environment. You can change these settings at any time after the initial installation. You might change them to accommodate changes in your site's library information, for example. If you change the information in the defaults.conf file after the system is running, you must then issue commands to propagate the defaults.conf file changes to the file system. For information, see Propagating Configuration File Changes to the System.

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

Example – Example **defaults.conf** File

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

Another sample file is located in /opt/SUNWsamfs/examples/defaults.conf.

How to Customize Default Values

1. Read the defaults.conf(4) man page to determine the defaults you want to change.
2. Use the cp(1) command to copy /opt/SUNWsamfs/examples/defaults.conf to its functional location.
   For example:

```
# cp /opt/SUNWsamfs/examples/defaults.conf /etc/opt/SUNWsamfs/defaults.conf
```

3. Use vi(1) or another editor to edit the file.
   Edit the lines that control those aspects of the system that you want to change. Remove the pound character (#) from column 1 of the lines you change.
4. Use the pkill(1M) command to send a SIGHUP signal to the sam-fsd(1M) daemon.
   For example:

```
# pkill -HUP sam-fsd
```

This command restarts the sam-fsd(1M) daemon and enables the daemon to recognize the changes in the defaults.conf file.

Features You Can Control From **defaults.conf**

This section describes two common features that you can control from the defaults.conf file. For more information, see the defaults.conf(4) man page.

Barcodes

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

Table – The **labels** Directive in the **defaults.conf** File

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

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

Drive Timing Values

You can set the unload and unload wait time for devices using the dev_unload and dev_delay directives, respectively. These directives enable you to set values that meet your site's requirements.

The format of the dev_unload parameter is as follows:

```
<dev>_unload = <seconds>
```

For dev, specify the device type as specified in the mcf(4) man page.

For seconds, specify the number of seconds that you want the system to wait after an unload command is issued. This gives the automated library time to eject the cartridge, open the door, and perform other operations before the cartridge is removed. The default is 0.

The format of the dev_delay directive is as follows:

```
<dev>_delay = <seconds>
```

For dev, specify the device type as specified in the mcf(4) man page.

For seconds, specify the minimum number of seconds that you want to have elapse between the time when a cartridge is loaded and the time when the same cartridge is able to be unloaded. The default is 30.

For example:

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

Configuring the Remote Notification Facility

The software can be configured to notify you when potential problems occur in its environment. The system sends notification messages to a management station of your choice. The Simple Network Management Protocol (SNMP) software within the software manages the exchange of information between network devices such as servers, automated libraries, and drives.

The Sun SAM Management Information Base (MIB) defines the types of problems, or events, that the Sun SAM software can detect. The software can detect errors in configuration, `tapealert`(1M) events, and other atypical system activity. For more information, see `/var/snmp/mib/SUN-SAM-MIB.mib`.

The following procedures describe how to enable and disable remote notification.

How to Enable Remote Notification

1. Ensure that the management station is configured and known to be operating correctly.
   Setting Up the Network Management Station describes this prerequisite.
2. Using `vi`(1) or another editor, examine the `/etc/hosts` file to ensure that the management station to which notifications should be sent is defined.
   The following sample file defines a management station with a host name of `mgmtconsole`.

```
999.9.9.9       localhost
999.999.9.999   loggerhost      loghost
999.999.9.998   mgmtconsole
999.999.9.9     samserver
```

3. Save your changes to `/etc/hosts` and exit the file.
4. Using `vi`(1) or another editor, open the file `/etc/opt/SUNWsamfs/scripts/sendtrap` and locate the `TRAP_DESTINATION=`hostname'` directive.
   This line specifies that remote notification messages be sent to port 161 of the server upon which the Sun Storage Archive Manager software is installed. Note the following:
   - If you want to change the host name or/and port, replace the `TRAP_DESTINATION` directive line with `TRAP_DESTINATION="`management-console-name`:`port`"`. Note the use of quotation marks (" ") rather than apostrophes (` ') in the new directive.
   - If you want to send remote notification messages to multiple hosts, specify the directive in the following format:

```
TRAP_DESTINATION="<mgmt-console-name>:<port> [ <mgmt-console-name>:<port>]"
```

   For example:

```
TRAP_DESTINATION="localhost:161 doodle:163 mgmt_station:1162"
```

5. Locate the `COMMUNITY="public"` directive in `/etc/opt/SUNWsamfs/scripts/sendtrap`.
   This line acts as a password. It prevents unauthorized viewing or use of SNMP trap messages. Examine this line and do one of the following, depending on the community string value of your management station:
   - If your management station's community string is also set to `public`, you do not have to edit this value.
   - If your management station's community string is set to a value other than `public`, edit the directive to replace `public` with the value that is used in your management station.
6. Save your changes to `/etc/opt/SUNWsamfs/scripts/sendtrap` and exit the file.

How to Disable Remote Notification

The remote notification facility is enabled by default. If you want to disable remote notification, perform this procedure.

1. If the file `/etc/opt/SUNWsamfs/defaults.conf` does not exist, use the `cp`(1) command to copy file `/opt/SUNWsamfs/examples/defaults.conf` to `/etc/opt/SUNWsamfs/defaults.conf`.
2. Open the file `/etc/opt/SUNWsamfs/defaults.conf` and find the line that specifies SNMP alerts. The line is as follows:

```
#alerts=on
```

3. Edit the line to disable SNMP alerts.
   Remove the # symbol and change `on` to `off`. After editing, the line is as follows:

```
alerts=off
```

4. Save your changes and exit the file.
5. Use the `pkill`(1M) command to send a SIGHUP signal to the `sam-fsd`(1M) daemon.

```
# pkill -HUP sam-fsd
```

   This command restarts the `sam-fsd`(1M) daemon and enables the daemon to recognize the changes in the `defaults.conf` file.

Adding the Administrator Group

By default, only the superuser can execute administrator commands. However, during installation you can supply an administrator group name. The `pkgadd`(1M) process prompts you for this group name during the installation.

Members of the administrator group can execute all administrator commands except for `star`(1M), `samfsck`(1M), `samgrowfs`(1M), `sammkfs`(1M), and `samd`(1M). The administrator commands are located in `/opt/SUNWsamfs/sbin`.

After installing the package, you can use the `set_admin`(1M) command to add or remove the administrator group. This action performs the same function as selecting an administrator group during the package installation. You must be logged in as superuser to use the `set_admin`(1M) command. You can also undo the effect of this selection and make the programs in `/opt/SUNWsamfs/sbin` executable only by the superuser. For more information, see the `set_admin`(1M) man page.

How to Add the Administrator Group

1. Choose a group name, or select a group that already exists within your environment.
2. Use the `groupadd`(1M) command, or edit the `/etc/group` file.
   The following is an entry from the group file designating an administrator group for the software. In this example, the `samadm` group consists of both the `adm` and `operator` users.

```
samadm::1999:adm,operator
```

How to Enable System Logging

The software logs errors, cautions, warnings, and other messages using the standard Sun Solaris `syslog`(3) interface. By default, the Sun SAM facility is `local7`.

1. Open the `/etc/syslog.conf` file.
2. From the file `/opt/SUNWsamfs/examples/syslog.conf_changes`, locate the logging line, which is similar, if not identical, to the following:

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

> Note -
> The preceding entry is all one line and has a TAB character (not a space) between the fields.

   The default facility is `local7`. If you set logging to something other than `local7` in the `/etc/syslog.conf` file, edit the `defaults.conf` file and reset it there, too. For more information, see the `defaults.conf`(4) man page.

3. Append the logging line from `/opt/SUNWsamfs/examples/syslog.conf_changes` to the `/etc/syslog.conf` file.
   For example:

```
# cp /etc/syslog.conf /etc/syslog.conf.orig
# cat /opt/SUNWsamfs/examples/syslog.conf_changes >> /etc/syslog.conf
```

4. Create an empty log file and send the `syslogd` a HUP signal.
   For example, to create a log file in `/var/adm/sam-log` and send the HUP to the `syslogd` daemon, type the following:

```
# touch /var/adm/sam-log
# pkill -HUP syslogd
```

   For more information, see the `syslog.conf`(4) and `syslogd`(1M) man pages.

5. (Optional) Use the `log_rotate.sh`(1M) command to enable log file rotation.
   Log files can become very large, and the `log_rotate.sh`(1M) command can help in managing log files. For more information, see the `log_rotate.sh`(1M) man page.

Configuring Other Sun Storage Products

The Sun SAM-QFS installation and configuration process is complete. You can configure related storage products at this time. For example, to configure the Sun SAM-Remote software, see Using the Sun SAM-Remote Software.

Related Topics

- Creating Parameters Files for Network Attached Automated Libraries
- Checking the Drive Order in Libraries
- Configuring the Archiver
- Managing Automated Libraries and Manually Loaded Drives
- Populating the Catalog

Installing SAM-QFS Manager

Contents

Installing and Configuring SAM-QFS Manager

If you want to use the browser user interface to configure and manage your SAM-QFS environment, follow the instructions in this section.

About SAM-QFS Manager

The SAM-QFS Manager is a browser interface tool that enables you to configure, control, protect, and monitor the archiving and file systems in your network from a central location. To access this central location, you can use the web browser on any host in your network.

The goal of the software is to provide a less complex way than command-line interface (CLI) commands to perform the most common archiving and file system tasks.

By default, SAM-QFS Manager is set up to manage the server on which it is installed. It can also be used to manage other servers running Sun Storage Archive Manager software, but those additional servers must first be configured to allow SAM-QFS Manager access. For instructions on how to add additional managed servers, see How to Add an Additional Server for SAM-QFS Manager Access.

Installing SAM-QFS Manager

Perform the tasks in this section to install the SAM-QFS Manager software.

Verifying Requirements for SAM-QFS Manager

You can install the SAM-QFS Manager software in one of the following configurations:

- As a stand-alone management station to manage one or more SAM-QFS Manager hosts
- As additional software on the SAM-QFS Manager host

After the SAM-QFS Manager software is installed, you can invoke SAM-QFS Manager from any machine on the network that is allowed access to its web server.

The host upon which you are configuring SAM-QFS Manager must meet the requirements described in the following sections.

Hardware Requirements

The minimum hardware requirements for the SAM-QFS Manager software are as follows:

- SPARC 400-MHz (or more) CPU or x64 AMD CPU
- 1 gigabyte of memory
- One 20-gigabyte disk
- At least 250 Mbytes free space in `/tmp`
- At least 100 Mbytes free space in / (the root partition)
- One 10/100/1000Base-T Ethernet port

Browser Requirements

Ensure that your installation meets the following browser requirements:

- One of the following browsers, at the minimum level indicated, must be used to access the File System Manager software:
  - Netscape™ 7.1 / Mozilla™ 1.7 / Firefox 1.5 on the Solaris OS or Microsoft Windows 98 SE, ME, 2000, or XP operating system
  - Internet Explorer 6.0 on the Microsoft Windows 98 SE, ME, 2000, or XP operating system
- You must enable JavaScript™ technology in your browser. Specific locations differ by browser. For example, in Firefox, go to the Content section of the Preferences panel and confirm that the box next to Enable JavaScript is checked.
- If you are upgrading from an earlier version of the SAM-QFS Manager software, you must clear the browser cache before using SAM-QFS Manager for the first time.

Operating System Requirements

Make sure that the following minimum level of the Solaris OS is installed on the web server:

- Solaris 10 Update 6

Web Software Requirements

The SAM-QFS Manager installation packages include revisions of the following software at the minimum levels indicated:

- Java™ 2 Standard Edition version 1.5.0
- JavaHelp™ 2.0
- Java Studio Enterprise Web Application Framework (JATO) 2.1.2
- Tomcat version 4.0.5

During the installation procedure, you will be asked to answer questions. Based on your answers, the installation software can install the correct revisions for you if the compatible revisions of these software packages are not present.

> **Note -**
> SAM-QFS Manager is registered in the Sun Java Web Console and can coexist with other applications that use the same console. The Java Web Console uses port 6789. This is an IANA-reserved port, so no application other than Java Web Console should use this port.

How to Install SAM-QFS Manager

Before You Begin

Ensure that you have met the installation requirements described in Verifying Requirements for SAM-QFS Manager.

Steps

1. Log in to the server that you want to use as the SAM-QFS management station.
   You can use the same server on which you installed the `SUNWsamfsr` and `SUNWsamfsu` packages, or you can use a different server on the same network.
2. Become superuser.
3. Go to the directory where the software package release files reside on your server.
4. Execute the `fsmgr_setup` script to start the installation process.
   For example:

   ```
   # fsmgr_setup
   ```

5. Answer the questions as prompted by the `fsmgr_setup` script.
   During the installation procedure, you are asked questions about your environment.
   The `fsmgr_setup` script automatically installs the following:
   - The `SUNWfsmgrr` package.
   - The `SUNWfsmgru` package.
     The installation script prompts you to specify whether you want to install localized packages.
     After installing the packages, the install software starts the Tomcat Web Server and enables logging.
6. Edit your system setup files to include the correct paths to commands and man pages.
   - In the Bourne or Korn shell, edit the `.profile` file, change the `PATH` and `MANPATH` variables, and export the variables.
     The following code example shows how your `.profile` file might look after editing.

     ```
     PATH=$PATH:/opt/SUNWfsmgr/bin
     MANPATH=$MANPATH:/opt/SUNWfsmgr/man
     export PATH MANPATH
     ```

   - In the C shell, edit the `.login` and `.cshrc` files.
     When you have finished editing, the `path` statement in your `.cshrc` file might look like the following line:

     ```
     set path = ($path /opt/SUNWfsmgr/bin)
     ```

     The following code example shows how the `MANPATH` in your `.login` file might after you have finished editing.

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

7. Use the `ps`(1) and `grep`(1) commands to make sure that the `rpcbind` service is running:

   ```
   # ps -ef | grep rpcbind
   ```

8. Examine the output from the preceding commands.

The output should contain a line similar to the following:

```
root    269      1  0    Feb 08 ?        0:06 /usr/sbin/rpcbind
```

If `rpcbind` does not appear in the output, type the following command to start the `rpcbind` service:

```
# /usr/sbin/rpcbind
```

9. (Optional) Start the SAM-QFS Manager (`fsmgmtd`) daemon.
   If you did not choose to start the SAM-QFS Manager daemon automatically during the installation process, do one of the following:
   - Type the following command to start the SAM-QFS Manager daemon and have it restart automatically every time the daemon process dies. With this configuration, the daemon also automatically restarts at system reboot.

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

   - Type the following command if you want the SAM-QFS Manager daemon to run only once and not automatically restart.

     ```
     # /opt/SUNWsamfs/sbin/fsmadm start
     ```

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

10. (Optional) Give additional users access to SAM-QFS Manager.
    By default, the `root` user has privileges to perform all operations available from the SAM-QFS software. You can assign other users full or partial access to SAM-QFS Manager operations.
    To give an additional user access to SAM-QFS Manager, use the `useradd` command. See How to Create Additional SAM-QFS User Accounts and How to Assign Privilege Levels to SAM-QFS Users for information about adding users and assigning SAM-QFS Manager user privilege levels.

How to Start SAM-QFS Manager

> Note -
> Before you start SAM-QFS Manager, disable all pop-up blockers.

1. Log in to the server where SAM-QFS Manager is installed, or log in to any system that has network access to it.
2. If you upgraded from a previous version of the software, open the web browser and clear the browser cache.
3. From the web browser, start SAM-QFS Manager.

   ```
   https://<hostname>:6789
   ```

   For hostname, type the name of the host where the SAM-QFS Manager software is installed. If you need to specify a domain name in addition to the host name, specify the hostname in this format: hostname`.`domainname. Note that this URL begins with `https`, not `http`.
   The Sun Java Web Console login page is displayed.

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

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

5. At the Password prompt, type the password.
6. Click Log In.
7. In the Storage section of the Applications page, select SAM-QFS Manager.

You are now logged in to SAM-QFS Manager.

Configuring SAM-QFS Manager

After SAM-QFS Manager is installed, you can log in to the software using the `root` user name and the password for the management station.

The `root` login grants you full administrator privileges to configure, monitor, control, and reconfigure the devices in your SAM-QFS Manager environment. Only the SAM-QFS Manager administrator should log in using the `root` login. All other users should log in using another user name.

By default, SAM-QFS Manager is set up to manage the server on which it is installed. It can also be used to manage other servers running SAM-QFS Manager software, but those additional servers must first be configured to allow SAM-QFS Manager access. For instructions on adding additional managed servers, see How to Add an Additional Server for SAM-QFS Manager Access.

How to Create Additional SAM-QFS User Accounts

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

If you remove the SAM-QFS Manager software, the removal scripts do not remove any additional accounts that you create manually. You must use one or both of the following procedures to administer any accounts you add manually.

Steps

1. Outside of the browser interface, log in to the SAM-QFS management station as `root`.
2. Use the `useradd` and `passwd` commands to add each user.
   For example, to add a user with account name `bobsmith`, type the following:

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

   Each user account that you add in this way has read-only viewing privileges for SAM-QFS Manager functions. To add additional privileges see How to Assign Privilege Levels to SAM-QFS Users.

How to Assign Privilege Levels to SAM-QFS Users

You can assign users full or partial access to SAM-QFS Manager functions.

1. Log in to the SAM-QFS management station as root.
2. To specify full or partial configuration privileges for a user, add the following line to the `/etc/user_attr` file.
   account-name**::::auths=**privilege-level
   - account-name is the name of the user's account.
   - privilege-level is the level of authorization to assign to the user.
     The following table lists the five levels of privileges you can assign to SAM-QFS Manager users.

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

Example – Assigning Full Privileges to a User

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

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

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

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

How to Create a SAM-QFS Manager Account for Multiple Users

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

1. Use the `useradd` and `passwd` commands to add the account.
   For example, to add a user account called `guest` for multiple users, type the following:

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

2. Use the `roleadd` and `passwd` commands to add the role.
   To create a role called `admin` with special privileges within the `guest` account, type the following:

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

3. Specify the privilege levels in the `/etc/user_attr` file.
   To assign the `admin` role privileges to restore and stage file systems, add the following lines to the `/etc/user_attr` file:

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

   In this example, when a user logs in as `guest`, SAM-QFS Manager prompts the user to select either No Role or Admin. If users know the Admin role password, they can select Admin, provide the Admin password, and have privileges to restore and stage file systems. All other users must select No Role and have read-only privileges.
   Because multiple users with the same privilege level can be logged in to the software concurrently, one user's changes could potentially overwrite another user's previous changes. To prevent this situation, develop policies about who can make changes and how to notify others.

How to Add an Additional Server for SAM-QFS Manager Access

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

1. Outside of the browser interface, use the `telnet` utility to connect to the server you want to add.
   Log in as `root`.
2. Use the `fsmadm`(1M) `add` command to add the SAM-QFS management station to the list of hosts that can remotely administer this server.
   Only hosts that are added to the list through this command can remotely administer the server.
   For example:

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

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

How to Set the SAM-QFS Manager Session Timeout

The Java Web Console framework has a default session timeout of 15 minutes. The SAM-QFS Manager installation program changes the session timeout to 60 minutes. You can change the session timeout to a different value, but to preserve security you should not set it to a value greater than 60 minutes.

1. To change the session timeout value, enter the following command on the SAM-QFS management station:

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

Example – Changing the Session Timeout Value

To change the timeout value to 45 minutes, you would type:

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

Uninstalling SAM-QFS Manager

Uninstalling the SAM-QFS Manager Software

For instructions on uninstalling the Sun QFS or SAM-QFS packages, see Removing the Existing Software.

How to Uninstall the SAM-QFS Manager Software

1. Log in to the server on which SAM-QFS Manager software is installed.
   This is the host on which you ran the `fsmgr_setup` script at installation time.
2. Become superuser.
3. Issue the following command to remove the SAM-QFS Manager software and all the applications that were installed with it:

```
# /var/sadm/samqfsui/fsmgr_uninstall
```

   This script prompts you to confirm removal of the Tomcat Web Server, JRE packages, and information pertaining to administrator and user accounts.

Upgrading Hardware

Upgrading Hardware

Contents

- Configuring EFI Labels for Shared x64 and SPARC Volumes

Preparing for a Hardware Device Upgrade

This section prepares you for hardware upgrades to devices within your environment.

General Prerequisites

Before starting the upgrade process, be sure to do the following:

- Determine whether the hardware addition or change requires a software upgrade from Sun Microsystems.
  Examples of changes that require a software upgrade include changes to the class of your server or significant increases in storage capacity. Examples of changes that do not require a software upgrade include
  additions to memory and increases in disk cache.

- If you are switching from a SPARC to an AMD server platform (or from AMD to SPARC), you must take precautions to prevent loss of data. See Switching Between SPARC and AMD Platforms for details.

- Read the hardware manufacturer's installation instructions carefully. Also read the information on adding hardware in your Solaris OS system administrator documentation.

- Check the Equipment Ordinal values in your old and new `mcf` files. For information about the `mcf` file, see the `mcf`(4) man page.

- Decide whether the backup copies you have on hand are sufficient. For information about backing up your data and metadata, see the procedures described in Setting Up Dump Files.
  - In a Sun QFS environment, the qfsdump(1M) command dumps all data and metadata. For more information about this process, see the qfsdump(1M) man page.
  - In SAM-QFS environments, the samfsdump(1M) command dumps all metadata. You must ensure that all files that need to be archived have an archive copy. Use the archive_audit(1) command on each SAM-QFS file system to see which files do not have an archive copy. In the following example, `/sam` is the mount point.

    ```
    # archive_audit /sam
    ```

- Ensure that the system is quiet, with no users logged in.

- In SAM-QFS environments, ensure that the archiver is in wait mode. The archiver must be in wait mode, and not running, during an upgrade.
  You can idle the archiver in one of the following ways:
  - Insert a wait directive into the `/etc/opt/SUNWsamfs/archiver.cmd` file. For more information about the wait directive and the `archiver.cmd` file, see the `archiver.cmd`(4) man page.
  - Use the `samu`(1M) operator utility.
  - Issue the following command:

    ```
    # samcmd aridle
    ```

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

Switching Between SPARC and AMD Platforms

The following are some important considerations if you are combining or changing between SPARC and x86 hardware platforms:

- Sun QFS software is supported only for the Solaris 10 OS on x64 platforms (AMD64 architecture), not for the EM64T architecture. With the exception of the Sun QFS shared Linux client and the Solaris 10 x86 shared client, it is also not supported for any 32-bit x86 architectures.

- All functions supported by Sun QFS software on the SPARC platform are also supported on the x64 platform except for the following:
  - The ADIC/Grau, Fujitsu LMF, IBM 3494, and Sony network attached libraries are not supported on x64 platforms. Sun StorageTek ACSLS-attached automated libraries are supported on x64 platforms.
  - Optical (MO and UDO) storage libraries and drives are not supported on x64 platforms.
  - SANergy software is not supported on x64 platforms.
  - SCSI-attached tape drives are not supported on x64 platforms because of a lack of support in the SCSI HBA 64-bit drivers for large block sizes. Both SCSI-attached libraries and fibre-attached libraries are supported with fibre-attached tape drives.
- EFI labels are required on all disks if your Sun QFS shared file system configuration contains both the Solaris 10 OS on x64 platforms

and the Solaris 10 OS on SPARC platforms. See Configuring EFI Labels for Shared x64 and SPARC Volumes for information on relabeling disks.

- Exercise caution when accessing the same SAN-attached storage from a Solaris environment on both SPARC and x64 platforms. The Solaris OS on x64 platforms cannot interpret the SMI VTOC8 disk label created by the Solaris OS on SPARC platforms, and the Solaris OS on SPARC platforms cannot interpret the SMI VTOC16 disk label created by the Solaris OS on x64. This can make it appear as though a disk is unlabeled, when in fact it is labeled and in use by a platform of a different architecture type. For example, a disk that is labeled with SMI VTOC8 may have mounted partitions in use by Solaris on a SPARC platform, but will appear as unlabeled when viewed with the `format`(1M) partition command by Solaris on an x64 platform. If you make the mistake of running `fdisk`(1M) as prompted by the `format`(1M) command, you will destroy the contents of that disk.
- You cannot change the architecture type of the server responsible for control of the file system metadata operations (that is, the server that was used to create the file system with the `sammkfs`(1M) command).
  - For a Sun QFS stand-alone file system, this means that you cannot mount the file system on a server that has a different architecture type from the one that created it.
  - For a Sun QFS shared file system, this means that you cannot change the architecture type of the metadata server or any potential metadata servers. This is because the different architectures use different byte-ordering schemes (endianness). However, you can migrate data from one architecture type to the other by copying the file system to temporary storage using either `qfsdump`(1M) or `samfsdump`(1M), re-creating the file system using `sammkfs`(1M), and then repopulating the file system with `qfsrestore`(1M) or `samfsrestore`(1M).
- The Sun StorageTek Traffic Manager I/O multipathing feature (MPxIO) is disabled by default for the Solaris 10 OS on the SPARC platform and enabled by default for the Solaris 10 OS on x64. This feature should be configured the same way for all systems in your Sun QFS shared file system configuration. It is configured in `/kernel/drv/fp.conf` for the Solaris 10 OS.
- In a Sun QFS shared file system environment, a configuration error will be generated if you have potential metadata servers of different architecture types (SPARC and x64) defined in the `/etc/opt/SUNWsamfs/hosts.fs` file.

Configuring EFI Labels for Shared x64 and SPARC Volumes

> Caution -
> Relabeling a disk will destroy the contents of that disk.

Use the Solaris prtvtoc(1M) command to determine whether a disk contains SMI or EFI labels. Under the Dimensions section of the output, SMI labels list the number of accessible cylinders, whereas EFI labels list the number of accessible sectors.

To convert disk labels from the default SMI VTOC8 to EFI, copy the file system to temporary storage using qfsdump(1M) or samfsdump(1M), relabel the disks with EFI labels using the format -e command, re-create the file system using sammkfs(1M), and repopulate the file system with qfsrestore(1M) or samfsrestore(1M).

When using the Solaris format -e command to create EFI labels, you can select the partition command from the menu to create and modify partitions (slices). When doing this, you must specify a tag id name of usr, rather than stand or unassigned, for EFI labels.

Note that EFI labels reserve the first 34 sectors, which misaligns Sun RAID-5 storage from a performance perspective. Unless you realign the storage, you will incur a RAID-5 read/modify/write performance penalty whenever writing. You can avoid this performance penalty by selecting the proper starting sector for all disk partitions for your particular storage configuration. For example, an 8+P Sun StorageTek T3 array with a 64K block size should have starting sectors that are multiples of 1024 for all disk slices `((8 * 64 * 1024) / 512 = 1024)`. Similarly, a 5+P Sun StorageTek 3510 FC array with a 128K block size should have starting sectors that are multiples of 1280 for all disk slices `((5 * 128 * 1024) / 512 = 1280)`.

Upgrading QFS and SAM-QFS

Contents

Upgrading to SAM-QFS 5.1

This section describes the procedures for upgrading a server to the 5.1 release of the Sun Storage Archive Manager or Sun QFS software. Use these procedures if you are upgrading your storage archive management functionality or Sun QFS file system.

> Note -
>
> - You must perform all the tasks in this section as superuser.
> - If you are upgrading from SAM-QFS 5.0 in a shared environment, you can perform a rolling upgrade. For more information, see Support for Rolling Upgrades in a Shared Environment.

Upgrade Overview Task Map

Depending on the the features that you need to support, you must complete several of the following procedures.

Step	Task	Description
1	Review the upgrade considerations.	Follow the established best practices.
2	Run the `samfsconfig(1m)` command.	Generate a current configuration report before you make changes.
3	Back up your existing file systems.	For existing QFS configurations, back up existing file system data and metadata. For existing archiving configurations, back up only the metadata.
4	Stop any archiving operations.	If you use the archiving features, stop all archiving before you continue with the upgrade.
5	Unshare any shared file systems.	If you have shared file systems, you must also unshare them.
6	Unmount the file systems.	Unmount all file systems.
7	(Optional) Perform necessary tasks to upgrade hardware devices.	This task is needed only if you need to upgrade hardware as well as Sun Storage Archive Manager or Sun QFS software. See Upgrading Hardware.
8	Remove existing software.	Use the `pkgrm(1m)` utility to remove the old software packages.
9	Add the new packages.	Similar to an initial install, use the `pkgadd(1m)` utility to add the packages.
10	(Optional) Upgrade SAM-QFS Manager.	If you used File System Manager, install the new SAM-QFS Manager package.

11	Restore the file systems.	Restore, reinitialize, verify, and remount the file systems.

Preparing for an Upgrade

Follow the instructions in this section to prepare for your upgrade.

Upgrade Considerations

When you decide to upgrade the host system being used for the file system, consider the following:

- Move to the new host while the existing host is still in operation. This practice enables you to install, configure, and test the new hardware platform with your applications.
- Moving to a new host system is equivalent to installing the Sun QFS software for the first time. In SAM-QFS archiving environments, you need to reinstall the software and update the configuration files. These files include the `mcf` file, the `/kernel/drv/st.conf` file, and the `/etc/opt/SUNWsamfs/inquiry.conf` file. In addition, you need to copy your existing `archiver.cmd` and `defaults.conf` files to the new system, configure system logging, and so on.
- Before powering down the old host system, decide whether the backup copies you have on hand are sufficient. You might want to capture new dump files to re-create the file system on the new server. For more information about creating a dump file, see Setting Up Dump Files.
- If you need to upgrade the Solaris OS to support the latest products, see Upgrading the Solaris OS.

Preserving Information for an Upgrade

If you are about to add or change disks, controllers, or other equipment in your environment, it can be difficult to correct or regenerate all the file system descriptions in the `mcf` file. The `samfsconfig` (1M) command can help you by generating information about your file system and file system components after you make these changes.

The `samfsconfig`(1M) command examines the devices you specify, determines whether any of them have Sun QFS superblocks on them, and writes this information to stdout. It uses information from the discovered superblocks and aggregates the devices into a format similar to an `mcf` file. You can save this format and edit it to re-create a damaged, missing, or incorrect `mcf` file.

The command can retrieve the Family Set number of the base device (the file system itself), the file system type (`ma` or `ms`), and whether the file system is a shared file system.

Irregularities are flagged with one of the following:

- A pound sign (#). This indicates incomplete family set information.
- A greater-than sign (>). This indicates that more than one device name refers to a particular file system element.

The following examples show output from the `samfsconfig`(1M) command.

Example 1

In this example, the system administrator has put a list of device names into a file. These device names were for devices that were not accounted for in the environment and that the system administrator therefore wanted to examine for family sets. The results displayed in the following code example show some old fragments of family sets and several complete instances.

Code Example 1 - Output From the **`samfsconfig`** (1M) Command

```
mn# samfsconfig -v *`cat /tmp/dev_files'*

Device `/dev/dsk/c5t10d0s0' has a SAM-FS superblock.
Couldn't open '/dev/dsk/c5t10d0s1': I/O error
Device '/dev/dsk/c5t10d0s3' has a SAM-FS superblock.
Device '/dev/dsk/c5t10d0s4' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t10d0s5' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t10d0s6' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t10d0s7' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t11d0s0' has a SAM-FS superblock.
Couldn't open '/dev/dsk/c5t11d0s1': I/O error
Device '/dev/dsk/c5t11d0s3' has a SAM-FS superblock.
Device '/dev/dsk/c5t11d0s4' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t11d0s5' doesn't have a SAM-FS superblock (SBLK).
```

```
Device '/dev/dsk/c5t11d0s6' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t11d0s7' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t12d0s0' has a SAM-FS superblock.
Couldn't open '/dev/dsk/c5t12d0s1': I/O error
Device '/dev/dsk/c5t12d0s3' has a SAM-FS superblock.
Device '/dev/dsk/c5t12d0s4' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t12d0s5' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t12d0s6' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t12d0s7' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t13d0s0' has a SAM-FS superblock.
Couldn't open '/dev/dsk/c5t13d0s1': I/O error
Device '/dev/dsk/c5t13d0s3' has a SAM-FS superblock.
Device '/dev/dsk/c5t13d0s4' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t13d0s5' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t13d0s6' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t13d0s7' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t8d0s0' has a SAM-FS superblock.
Device '/dev/dsk/c5t8d0s1' has a SAM-FS superblock.
Device '/dev/dsk/c5t8d0s3' has a SAM-FS superblock.
Device '/dev/dsk/c5t8d0s4' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t8d0s5' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t8d0s6' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t8d0s7' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t9d0s0' has a SAM-FS superblock.
Couldn't open '/dev/dsk/c5t9d0s1': I/O error
Device '/dev/dsk/c5t9d0s3' has a SAM-FS superblock.
Device '/dev/dsk/c5t9d0s4' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t9d0s5' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t9d0s6' doesn't have a SAM-FS superblock (SBLK).
Device '/dev/dsk/c5t9d0s7' doesn't have a SAM-FS superblock (SBLK).
13 SAM-FS devices found.

#
# Family Set 'qfs1' Created Mon Jun 25 10:37:52 2004
#
# Missing slices
# Ordinal 0
# /dev/dsk/c5t8d0s1 10 mm qfs1 -
#
# Family Set 'qfs1' Created Wed Jul 11 08:47:38 2004
#

qfs1 200 ma qfs1 - shared

/dev/dsk/c5t8d0s3 201   mm  qfs1 -
/dev/dsk/c5t9d0s3 202  mr qfs1   -
/dev/dsk/c5t10d0s3 203  mr qfs1  -
/dev/dsk/c5t11d0s3 204  mr qfs1  -
/dev/dsk/c5t12d0s3 205  mr qfs1  -
/dev/dsk/c5t13d0s3 206  mr qfs1  -

#
# Family Set `sqfs1' Created Wed Nov  7 16:55:19 2004
#

sqfs1 100 ma sqfs1 - shared

/dev/dsk/c5t8d0s0 101   mm  sqfs1  -
/dev/dsk/c5t9d0s0 102  mr sqfs1  -
/dev/dsk/c5t10d0s0 103  g0 sqfs1  -
/dev/dsk/c5t11d0s0 104  g0 sqfs1  -
/dev/dsk/c5t12d0s0 105  g1 sqfs1  -
/dev/dsk/c5t13d0s0 106  g1 sqfs1  -
```

```
#
```

Example 2

In the output shown in the following code example, the devices flagged with a greater-than sign (>) are duplicated. The s0 slice starts at the start of disk, as does the whole disk (s2) slice. This is the style of output obtained in a Solaris 9 OS.

Code Example 2 - Output from the **samfsconfig** Command

```
# samfsconfig /dev/dsk/c3t*
#
# Family Set 'shsam1' Created Wed Oct 17 14:57:29 2001
#

shsam1 160 ma shsam1 shared
> /dev/dsk/c3t50020F23000055A8d0s2    161    mm   shsam1  -
> /dev/dsk/c3t50020F23000055A8d0s0    161    mm   shsam1  -
> /dev/dsk/c3t50020F23000055A8d0s1    162    mr   shsam1  -
> /dev/dsk/c3t50020F23000078F1d0s0    163    mr   shsam1  -
> /dev/dsk/c3t50020F23000078F1d0s2    163    mr   shsam1  -
> /dev/dsk/c3t50020F23000078F1d0s1    164    mr   shsam1  -
```

Backing Up Existing File Systems

You should back up your existing file systems before you upgrade the software. It is especially important to back up your existing file systems if the following conditions exist:

- You are currently using a version 1 superblock with a Sun StorageTek QFS 4U0 system and you want to reinitialize your file systems with a version 2A superblock. In To Reinitialize and Restore the File System, you reinitialize the file systems and restore your data.
- You suspect that your current `qfsdump`(1M) file is incorrect or outdated.

The following subsections explain the differences between the superblock versions and present the procedure for backing up your file systems:

- Using the Version 1 and Version 2 Superblocks
- How to Back Up a File System

Example - Using samfsinfo(1M) to Retrieve File System Information

The following example shows the `samfsinfo`(1M) command you use to retrieve information about the `qfs2` file system. The second line of output indicates that this file system is using a version 2 superblock.

```
# samfsinfo qfs2

samfsinfo: filesystem qfs2 is mounted.
name: qfs2       version:     2    shared
time:      Sun Sep 28 08:20:11 2003
count:     3
capacity:      05aa8000          DAU:         64
space:         0405ba00
meta capacity: 00b4bd20          meta DAU:    16
meta space:    00b054c0
ord  eq   capacity      space   device
  0  21   00b4bd20   00b054c0   /dev/md/dsk/d0
  1  22   02d54000   01f43d80   /dev/dsk/c9t50020F2300010D6Cd0s6
  2  23   02d54000   02117c80   /dev/dsk/c9t50020F2300010570d0s6
```

Using the Version 1, Version 2, and Version 2A Superblocks

By default, Sun QFS and SAM-QFS 5.0 create new file systems that have a version 2A superblock. This enables the following features:

- Large host table (larger than 16k bytes)

- Online grow using `samadm eq-add` or `samu add` command

A version 2A file system is not compatible with earlier product versions. For example, you cannot mount a version 2A file system on SAM-QFS 4.6. If you need to create a file system that you can mount on earlier releases of the product, use the `sammkfs -P` command.

> Note -
> If you use the `sammkfs -P` command to create a file system, that file system will not support either a large host table or online grow.

To support a large host table or online grow on an existing version 2 file system, use the `samfsck -u 2A` file-system command to upgrade the file system to version 2A.

> Caution -
> Adding features to the file system is not backwards compatible to any version of SAM-QFS previous to 5.0, and is not reversible.

How to Back Up a File System (QFS Configurations)

Follow these steps for each file system in your environment.

> Note -
> If you are upgrading from an existing archiving environment, you only have to back up the file system metadata. See How to Back Up a File System (SAM-QFS Configurations).

Steps

1. Become superuser from a console connection.
   If you have not already logged in as root, do so now.
2. Use the `boot`(1M) command to boot the system in single-user mode:

```
# boot -s
```

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

```
# mount /qfs1
```

4. Use the `qfsdump`(1M) command to back up the file data and metadata of the file system.
   The `qfsdump`(1M) command dumps file names, inode information, and file data. The destination of the `qfsdump`(1M) output (generally a file) must be at least as large as the file system that you are backing up. The destination location (disk or tape) must have enough space to hold the amount of file data and metadata that you are dumping. For more information about using the `qfsdump`(1M) command, see Setting Up Dump Files or see the `qfsdump`(1M) man page.
   Dump each file system to a location outside of the existing file system. For more information, see the `qfsdump`(1M) man page.
   For example, if you have a file system named `qfs1` (mounted at `/qfs1`) that you want to back up, your choices are as follows:
   - You can write the `qfsdump`(1M) output to a tape device.
     The following example shows how to write to a tape in device `/dev/rmt/1cbn`.

```
# cd /qfs1
# qfsdump -f /dev/rmt/1cbn
```

   - You can write the `qfsdump`(1M) output to a file in a UNIX file system (UFS).
     The following example shows how to write to a file in a UFS.

```
# cd /qfs1
# qfsdump -f /save/qfs/qfs1.bak
```

- You can initialize a new Sun QFS file system, and perform the qfsrestore(1M) command directly in that new file system. This alternative is applicable only if you have already installed the Sun QFS 5.0 software somewhere in your environment. For example, assume that you want to write the dump file into a second file system called qfs2 (mounted at /qfs2) and that you initialized the qfs2 file system using the 5.0 software. The following example shows how to accomplish this using commands.

```
# mount /qfs2
# cd /qfs1
# qfsdump -f - | (cd /qfs2; qfsrestore -f -)
```

For more information about backing up your file systems, see Setting Up Dump Files.

How to Back Up a File System (SAM-QFS Configurations)

Follow these steps for each file system in your environment.

> Note -
> If you are upgrading from an existing QFS environment, you need to back up the file system data and metadata. See How to Back Up a File System (QFS Configurations).

Steps

1. Become superuser from a console connection.
   If you have not already logged in as root, do so now.
2. Make sure that all files are archived.
   The following example assumes that sam1 is the mount point of the file system. You can complete this step by entering a command similar to the following:

   ```
   # sfind /sam1 ! -type d ! -archived > /tmp/notarchived.list
   ```

   The preceding command finds all files that are not archived and sends the output to a file. Depending on the size of your file system, this command can take a long time to complete.
3. Examine this command's output, and do one of the following:
   - If you want any of these unarchived files to appear in the dump file, archive them now.
   - Use the -u option to the samfsdump(1M) command to dump unarchived data if you suspect that some files have not yet been archived. The -u option can create very large dump files, however, so consider space limitations when using this option.
4. Use the samfsdump(1M) command to back up each file system's metadata. See About Recovery Points.
   Dump each file system's metadata to a location outside the existing file system.
   The following example assumes that you have a file system mounted at /sam1 that you want to back up to samfs1.dump, which exists outside of the current file systems.

Example – Using samfsdump (1M)

```
# cd /sam1
# samfsdump -f /csd_dump_dir/samfs1.dump
```

The samfsdump(1M) command dumps file names and inode information, not data. For more information, see the samfsdump(1M) man page.

You must back up the metadata information for each file system, so repeat the preceding steps for each file system in your SAM-QFS archiving environment.

For more information about backing up your file systems, see SAM-QFS Troubleshooting.

Stopping Archiving Operations

In existing SAM-QFS environments, you must stop all archiving operations before you unmount the file systems.

How to Stop Archiving Operations From SAM-QFS Manager

You can idle or stop archiving activity from the SAM-QFS Manager software.

1. From the Servers page, click the name of the server for which you want to control archiving activity.
   The File Systems Summary page is displayed.
2. Click the Archive Administration node in the navigation tree and click the Archive Activity sub-node.
   The Activity Management page is displayed.
3. In the Archiving section of the page, select the radio button next to the option that you want to control, and click Submit Job:
   - Idle - Stops archiving after it reaches a logical point in the process
   - Stop - Immediately stops the archiving process
4. To view the progress of the archiving activity, click the Jobs sub-node under the System Administration node in the navigation tree and look for the job that you submitted.
   For complete information about controlling archiving activity from SAM-QFS Manager, see the SAM-QFS Manager online help.

How to Stop Archiving Operations From the Command Line

1. Use the samcmd(1M) idle command to idle each removable media drive configured in your mcf file.
   This step enables the archiver, stager, and other processes to complete current operations. Use this command in the following format:

```
samcmd idle <eq>
```

   For eq, specify the equipment number of the device as defined in your `mcf` file.
   You can also idle the drives by using the `samu`(1M) operator utility. For more information about the samcmd(1M) command, see the samcmd(1M) man page.

2. If you are upgrading from a Sun StorageTek SAM 4U0 system, use the `samcmd`(1M) `aridle` command to idle the archiver:

```
# samcmd aridle
```

3. Use the `samd`(1M) `stop` command to stop all operations.
   For example:

```
# samd stop
```

Unsharing File Systems

If your file systems are NFS shared file systems, use the `unshare`(1M) command on the file system.

For example, the following command unshares the `qfs1` file system:

```
# unshare /qfs1
```

Unmounting File Systems

You can unmount a file system using any of the following methods described in this section. After the file system is unmounted, you can proceed to Removing Existing Sun QFS Software.

> Note -
> To unmount a shared file system, follow the instructions in the How to Unmount a Shared File System.

How to Unmount a File System Using SAM-QFS Manager

1. From the Servers menu, select the name of the server on which the file system is located.
   The File System Summary page is displayed.
2. Select the radio button next to the file system that you want to unmount.
3. From the Operations menu, choose Unmount.

How to Unmount Using CLI Commands

- Use the `umount(1M)` command to unmount each Sun QFS file system.
  - If necessary, use the `-f` option to the umount(1M) command to force the file system to unmount.
  - If `umount(1M)` is not successful, it might be because files in the file system are being used or because you have used the `cd` command to change to a directory that is within the file system. In this case, follow these steps:
  1. Use the `fuser(1M)` command to determine whether any processes are still busy.
     For example, the following command queries the `qfs1` file system:

```
# fuser -uc /qfs1
```

  2. If any processes are still busy, use the `kill(1M)` command to terminate them.
  3. Use the `umount(1M)` command to unmount each Sun QFS file system.

How to Unmount by Editing the /etc/vfstab File and Rebooting

1. Edit the `/etc/vfstab` file.
   For all file systems, change the Mount at Boot field from yes or delay to no.
2. Reboot the system.

Removing Existing Software

Use the `pkgrm(1M)` command to remove the existing software. You must remove any existing Sun Storage Archive Manager or Sun QFS packages before installing new packages.

If you are using any optional packages, make sure that you remove these packages before removing the main packages. The installation script prompts you to confirm several of the removal steps.

> Note -
> The `samu` utility must be closed before you remove the software.

How to Remove Existing Software

1. Use the `pkginfo(1)` command to determine which software packages are installed on your system.
   For example:

```
# pkginfo | grep qfs
# pkginfo | grep sam
```

2. Use the `pkgrm(1M)` command to remove the existing packages.
   The following example command removes the SUNWqfsu and the SUNWqfsr packages from a 4U1 release:

```
# pkgrm SUNWqfsu SUNWqfsr
```

> Note -
> The SUNWqfsr package must be the last package removed. The 4U1 release does not include any localized software packages.

The following example command removes the SUNWcqfs, the SUNWfqfs, and the SUNWjqfs localized packages from a 4U0 release:

```
# pkgrm SUNWcqfs SUNWfqfs SUNWjqfs SUNWqfs
```

Note -
The SUNWqfs package must be the last package removed.

Adding the Upgrade Packages

The software packages use the Sun Solaris packaging utilities for adding and deleting software. The `pkgadd`(1M) command prompts you to confirm various actions necessary to upgrade the packages.

During the installation, the system detects the presence of conflicting files and prompts you to indicate whether you want to continue with the installation. You can go to another window and copy the files that you want to save to an alternate location.

How to Add the Packages

1. Use the `cd`(1) command to change to the directory where the software package release files reside.
   This is one of the following, depending on your release media:
   - If you downloaded the release files as described in Obtaining the Release Files, change to the directory to which you downloaded the files.
   - If you obtained the release files from a CD-ROM, change to the directory on the CD-ROM that corresponds to your OS version.
2. Use the `pkgadd`(1M) command to upgrade the packages.
   For example:

```
# pkgadd -d . SUNWqfsr SUNWqfsu
```

3. Enter yes or y in response to each of the questions.
   During the installation, the system detects the presence of conflicting files and prompts you to indicate whether or not you want to continue with the installation. You can go to another window and copy any files you want to save to an alternate location.

Upgrading SAM-QFS Manager

To upgrade SAM-QFS Manager, just install the new SAM-QFS Manager package. The installation process informs you if a previous version of the product exists and takes the appropriate steps to upgrade it. For information, see Installing SAM-QFS Manager.

Restoring the File System

The instructions in this section cover the tasks involved in restoring the file system after an upgrade.

How to Verify the mcf File

1. Type the `sam-fsd`(1M) command.
2. Examine the output for errors, as follows:
   - If the `mcf` file is free of syntax errors, the `sam-fsd`(1M) output is similar to that shown in the following example. The output contains information about the file systems and other system information.
     `sam-fsd` (1M) Output Showing No Errors

```
# sam-fsd
 Trace file controls:
 sam-amld      off
 sam-archiverd off
 sam-catserverd off
 sam-fsd       off
 sam-rftd      off
 sam-recycler  off
 sam-sharefsd  off
 sam-stagerd   off
 sam-serverd   off
 sam-clientd   off
 sam-mgmt      off
```

- If the `mcf` file contains syntax or other errors, the errors are shown in the output.
  If your mcf file has errors, see Setting Up the Environment Configuration and the `mcf`(4) man page for information about how to create this file correctly.

> Note -
> If you change the `mcf` file after the Sun QFS file system is in use, you must convey the new `mcf` specifications to the Sun QFS software. For information about propagating `mcf` file changes to the system, see the Sun QFS File System Configuration and Administration Guide.

How to Modify the /etc/vfstab File

Perform this task if you modified the `/etc/vfstab` file in Unmounting File Systems.

1. Edit the `/etc/vfstab` file again.
2. Change the Mount at Boot field for all Sun QFS file systems from no to yes or delay.

How to Reinitialize and Restore the File System

In this task, you reinitialize the file systems and restore the saved data in the new file systems. This task completes the process initiated in Backing Up Existing File Systems. To accomplish this, use the `sammkfs`(1M) and `qfsrestore`(1M) commands on each file system.

> Caution -
> The Sun QFS 4U2 and later software does not enable you to initialize a file system with a version 1 superblock. The Sun QFS 4U2 file system allows file systems to be initialized only with the version 2 superblock. If you are upgrading from 4U0 using a version 1 superblock, be aware that issuing a 4U2 or later sammkfs(1M) command at this point reinitializes your file system with a version 2 superblock.

1. Issue the `samfsinfo`(1M) command and examine the output.
   The output tells you the DAU size that was specified with the `sammkfs`(1M) command when the file system was created. You will use this DAU size again in the next step.
2. Use the `sammkfs`(1M) command to initialize a new Sun QFS file system.
   The following example command reinitializes a file system named `qfs1` with a DAU size of 512 kilobytes:

   ```
   # sammkfs -a 512 qfs1
   ```

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

3. Use the `qfsrestore`(1M) command to restore the dumped data in the new file system.
   For example, suppose you had a file system named `qfs1` (mounted at `/qfs1`) that you wanted to restore from files dumped to `qfs1.bak`, which existed outside of the Sun QFS file system. In this case, you would issue the following commands:

```
# cd /qfs1
# qfsrestore -f /save/qfs/qfs1.bak
```

How to Check the File System

Perform this task if you did not reinitialize and restore the file system as just described.

- Use the samfsck(1M) command to check each existing file system for inconsistencies.
  For more information, see the samfsck(1M) man page.

Mounting the File System

You can mount the file system using SAM-QFS Manager or the CLI.

> Note -
> For any file systems that are configured for archiving, archiving operations restart when the file system is mounted.

How to Mount the File System Using SAM-QFS Manager

1. From the Servers menu, select the name of the server on which the file system is located.
   The File System Summary page is displayed.
2. Select the radio button next to the file system that you want to mount.
3. From the Operations menu, choose Mount.

How to Mount the File System Using the CLI

- Issue the mount(1M) command.
  In the following example, qfs1 is the name of the file system to be mounted:

```
# mount qfs1
```

Recompiling API-Dependent Applications

File headers, the calling sequence, and other elements of the Sun QFS application programming interface (API) can change from release to release. If you are running applications that use the API, you should recompile them all at this time.

> Caution -
> Failure to recompile API-dependent applications at this point can cause your applications to generate unexpected results.

Related Topics

- Upgrading Hardware
- Upgrading the Solaris OS

Next Steps

Upgrading the Solaris OS

Upgrading the Solaris OS

Note -
You must perform all the tasks in this section as superuser.

Contents

The following section describes how to upgrade the Solaris OS when running the Sun QFS software.

How to Upgrade the Solaris OS in a Sun QFS Environment

Many of the steps involved in upgrading your Solaris OS level are identical to the steps involved in upgrading your Sun QFS environment. Some of the steps in this procedure reference procedures in the previous sections.

Note -
The SAM-QFS 5.0 software requires at least Solaris 10 Update 6 of the OS software.

1. Obtain the Sun QFS and Solaris OS software upgrades.
   Sun QFS software supports various levels of the Solaris OS. Do not reinstall your old Sun QFS software on your newly upgraded Solaris OS unless you are sure they are compatible.
   Contact your application service provider or Sun Microsystems to obtain new copies of the software.
2. Back up all site-customized system files and configuration files.
   These files include `mcf`, `defaults.conf`, `samfs.cmd`, the shared hosts files, and so on. Back up these files for all file systems in your Sun QFS environment. Also make sure that you have backup copies of files in the `/etc/opt/SUNWsamfs` directory.
3. Ensure that each affected file system is backed up.
   The file systems should be backed up regularly according to your site's policies and as described in Backing Up SAM-QFS Data and Files. If you are comfortable with the backup files that already exist for your file systems, there is no need to back them up again now.
4. Unmount the file systems.
   For instructions, see Unmounting File Systems.
5. Remove the existing Sun QFS software.
   You must remove the existing Sun QFS package before installing either the new package or the new operating system level. For instructions, see Removing Existing Software.
6. Upgrade the Solaris OS.
   Install the new Solaris OS revision using the corresponding Sun Solaris upgrade procedures.
7. Add the upgrade packages that you obtained in Step 1.
   The Sun QFS software package uses the Solaris OS packaging utilities for adding and deleting software. You must be logged in as superuser to make changes to software packages. The `pkgadd`(1M) command prompts you to confirm various actions necessary to upgrade the Sun QFS package. For instructions, see Adding the Upgrade Packages.
8. (Optional) Update the `mcf` file.
   If device names have changed, you might need to update the `mcf` file to match the new device names. Verify the new device names, and then follow the procedure in Restoring the File System.
9. If your `/etc/vfstab` file does not have yes in the Mount at Boot field, mount the file systems.
   Use the procedure described in Mounting the File System.

Related Topics

- Upgrading QFS and SAM-QFS
- Upgrading Hardware

Next Steps

Complete (Printable) SAM-QFS 5.1 Installation Guide

Sun Storage Archive Manager 5.1 Installation Guide

Contents

Preparing for Installation

This section explains the system requirements for the Sun QFS and SAM-QFS products and the tasks you must complete before you begin to install and configure your software.

Hardware and Software Requirements

You can install the software either on a Sun server based on UltraSPARC ® technology or on a server based on AMD Opteron x64 technology.

Additional requirements for the server that you want to use as the web server host for the SAM-QFS Manager browser interface tool are described in SAM-QFS Manager Requirements.

The software package runs on many Sun workstations and servers. Before installation, you should verify the compatibility of the hardware and the level of the Solaris Operating System (OS).

Operating System Requirements

Before installation, verify the applicability of the hardware and the level of the operating system. To install the software, you also must have root-level access to your system.

Sun Storage Archive Manager and Sun QFS 5.1 software require the following minimum operating system release:

- Solaris 10, Update 6

In addition, you can use any of the following operating systems as a client in a shared file system:

- Solaris 10 OS for x86 (32-bit)
- Red Hat Enterprise Linux 4.0 (UD-4) for x64 platforms

- Red Hat Enterprise Linux 4.5 for x64 platforms
- SuSE Linux Enterprise Server 9 (service pack 2) for x64 platforms
- SuSE Linux Enterprise Server 10 for x64 platforms
- SuSE Linux Enterprise Server 10 (service pack 2) for x64 platforms

How to Verify the Environment

Repeat these steps for each host on which you want to install the software.

1. Verify that your system has a CD-ROM drive or that it can access the release package at the Sun Download Center
2. Log in to your system as `root`.
   You must have superuser access to install the software.
3. Verify your system's Solaris OS level.
   For example, the output from the following command will show the major and minor OS release information as well as the architecture:

   ```
   % cat /etc/release
   ```

   The software relies on properly configured Solaris software at the following minimum release level:

   - Solaris 10, Update 6

Installing Solaris OS Patches

Sun Microsystems provides Solaris OS patches to customers with a maintenance contract by means of CD-ROM, anonymous FTP, and the Sun Microsystems SunSolve™ web site .

To install a patch after you install the Sun QFS or SAM-QFS release packages, load 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 included in the patch or jumbo patch cluster.

Software Host Requirements

If you plan to install the software in a multihost environment, such as a SAM-Remote configuration, all host systems must have the same software release level installed and operational.

Verifying Shared File System Requirements

This section describes the system requirements for a Sun QFS shared file system.

Metadata Server Requirement

You must have at least one Solaris metadata server. If you want to be able to change the metadata server, you must have at least one other Solaris host that can become the metadata server. These additional host systems are known as potential metadata servers. These servers must all be running on the same hardware platform, either SPARC or x64. You cannot mix server hardware platforms. In a Sun Cluster environment, all nodes included in a shared file system are potential metadata servers.

The following are configuration recommendations with regard to metadata storage:

- A shared file system should have multiple metadata (`mm`) partitions. This spreads out metadata I/O and improves file system throughput.
- A shared file system should use a separate, private metadata network so that typical user traffic does not interfere with metadata traffic. A switch-based (not hub-based) network is recommended.

Operating System and Hardware Requirements

Ensure that your configuration meets the following operating system and hardware requirements:

- The host systems to be configured in the Sun QFS shared file system must be connected by a network.
- All metadata servers and potential metadata servers must have the same processor type.
- The client systems can be installed on the Solaris OS or on one of the following operating systems:
  - Red Hat Enterprise Linux 4.0 (UD-4) for x64 platforms (Sun QFS shared client only)
  - Red Hat Enterprise Linux 4.5 for x64 platforms (Sun QFS shared client only)

- SuSE Linux Enterprise Server 9 (service pack 2) for x64 platforms (Sun QFS shared client only)
    - SuSE Linux Enterprise Server 10 for x64 platforms (Sun QFS shared client only)
    - SuSE Linux Enterprise Server 10 (service pack 2) for x64 platforms (Sun QFS shared client only)
- Online data storage devices must be directly accessible to all hosts. All online metadata storage devices must be directly accessible to all potential metadata server hosts.

Sun Storage Archive Manager and Sun QFS Release Levels

Ensure that your configuration meets the following requirements:

- Each host to be configured in the shared file system must have the same software package installed.
- All software installed on the systems in the shared file system must be at the same release level. For example, if one host has the SAM-QFS 5.1 packages, all hosts that are part of the shared file system must have the SAM-QFS 5.1 packages.
  This requirement ensures that all systems in a shared file system have identical over-the-wire protocol versions. If these levels do not match, the system writes the following message to the metadata server's `/var/adm/messages` file when mounting is attempted:

  ```
  SAM-FS: <client> client package version <x> mismatch, should be <y>.
  ```

- When applying patches or upgrading the software for a shared file system, make sure to apply the same patch to all hosts that have access to the shared file system. Unexpected results might occur if not all host systems are running the same patch level.

Verifying Third-Party Compatibilities

The SAM-QFS software interoperates with many different hardware and software products from third-party vendors. Depending on your environment, you might need to upgrade other software or firmware before installing the SAM-QFS package. Consult the Release Notes for information pertaining to library model numbers, firmware levels, and other compatibility information.

SAM-QFS Manager Requirements

The SAM-QFS Manager browser interface is used to configure, control, monitor, or reconfigure a SAM-QFS environment using a graphical web browser interface.

You can install the SAM-QFS Manager software in one of the following configurations:

- As a stand-alone management station to manage one or more Sun QFS hosts
- As additional software on the Sun QFS host

After the SAM-QFS Manager software is installed, you can invoke the SAM-QFS Manager from any machine on the network that is allowed access to its web server.

For information about the requirements for the host upon which you are configuring the SAM-QFS Manager software, see Verifying Requirements for SAM-QFS Manager.

Determining Disk Space Requirements

The SAM-QFS software packages require a certain amount of disk cache (file system devices) to create and manage data files and directories.

Planning Your File System and Verifying Disk Cache

A local file system requires only a single partition. If you install SAM-QFS to enable archiving support, the file system requires either one or two partitions:

- To store file data separately from file system metadata (`ma` file system), you need to have at least two disk devices or partitions.
- To store data and metadata on the same device (`ms` file system), you need to have one disk device or partition.

The disk devices or partitions do not require any special formatting. You might see better performance if you configure multiple devices across multiple interfaces (HBAs) and disk controllers.

> Caution -
> Make sure that the disks and partitions that you plan to use are not currently in use and do not contain any existing data. Any existing data will be lost when you create the Sun QFS file system.

The disks must be connected to the server through a Fibre Channel (FC) or SCSI controller. You can specify individual disk partitions for a disk, or you can use the entire disk as a disk cache. The software supports disk arrays, including those under the control of volume management software, such as Solaris Volume Manager.

Before creating your first file system, you should familiarize yourself with file system layout possibilities. For information about volume management, file system layout, and other aspects of file system design, see the Sun QFS File System Configuration and Administration Guide.

> Note -
> If you are using a shared file system configuration that contains the Solaris 10 OS on both x64 platforms and SPARC platforms, Extensible Firmware Interface (EFI) labels are required on all shared disks. See Configuring EFI Labels for Shared x64 and SPARC Volumes for information about relabeling disks.

How to Estimate Disk Cache Requirements

Use the following guidelines to estimate the disk cache needed for SAM-QFS software (file systems plus the storage and archive manager):

- Disk cache = largest file (in bytes) + amount of space needed for working files
- Metadata cache
  Use the following information to estimate the metadata cache requirements. The metadata cache must have enough space to contain the following data:
  - Two copies of the superblock (16 Kbytes each)
  - Reservation maps for metadata space plus data space((metadata + file data)/disk allocation unit (DAU)/32,000) * 4 Kbytes
  - Inode space(number of files + number of directories) * 512 bytes
  - Indirect blocks – a minimum of 16 Kbytes each
  - Directory data space(number of directories * 16 Kbytes)

Run the `format`(1M) command to verify that you have sufficient disk cache space.

The `format`(1M) command shows how the disks are partitioned and the size of each partition.

Example 1 - Using the format(1M) Command on Fibre-Channel-Attached Disks

This example shows six disks attached to a server. Two internal disks are connected by means of controller 0 on targets 10 and 11 (`c0t10d0` and `c0t11d0`). The other disks are external.

For the sake of clarity, the `format`(1M) command output in this example has been edited.

Example – **`format`**(1M) Command for Fibre-Channel-Attached Disks

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
       0. c0t10d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /sbus@3,0/SUNW,fas@3,8800000/sd@a,0
       1. c0t11d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /sbus@3,0/SUNW,fas@3,8800000/sd@b,0
       2. c9t60020F2000003A4C3ED20F150000DB7Ad0 <SUN-T300-0118 cyl 34530 alt 2 hd 48 sec 128>
          /scsi_vhci/ssd@g60020f2000003a4c3ed20f150000db7a
       3. c9t60020F2000003A4C3ED215D60001CF52d0 <SUN-T300-0118 cyl 34530 alt 2 hd 48 sec 128>
          /scsi_vhci/ssd@g60020f2000003a4c3ed215d60001cf52
       4. c9t60020F2000003A4C3ED21628000EE5A6d0 <SUN-T300-0118 cyl 34530 alt 2 hd 48 sec 128>
          /scsi_vhci/ssd@g60020f2000003a4c3ed21628000ee5a6
       5. c9t60020F2000003A4C3ED216500009D48Ad0 <SUN-T300-0118 cyl 34530 alt 2 hd 48 sec 128>
          /scsi_vhci/ssd@g60020f2000003a4c3ed216500009d48a
Specify disk (enter its number):^d
#
# format /dev/rdsk/c9t60020F2000003A4C3ED216500009D48Ad0s2
# *format f*
partition> p


Part      Tag    Flag     Cylinders         Size            Blocks
  0 unassigned    wm       0 -  4778       14.00GB    (4779/0/0)   29362176
  1 unassigned    wm    4779 -  9557       14.00GB    (4779/0/0)   29362176
  2     backup    wu       0 - 34529      101.16GB    (34530/0/0) 212152320
  3 unassigned    wm    9558 - 14336       14.00GB    (4779/0/0)   29362176
  4 unassigned    wm   14337 - 19115       14.00GB    (4779/0/0)   29362176
  5 unassigned    wm   19116 - 23894       14.00GB    (4779/0/0)   29362176
  6 unassigned    wm   23895 - 28673       14.00GB    (4779/0/0)   29362176
  7 unassigned    wm   28674 - 33452       14.00GB    (4779/0/0)   29362176

partition> ^D
#
```

Example 2 - Using the format(1M) Command on SCSI-Attached Disks

The following example shows four disks attached to a server. Two internal disks are connected by means of controller `0` on targets `0` (`c0t0d0`) and `1` (`c0t1d0`). Two external disks are connected by means of controller `3` on targets `0` (`c3t0d0`) and `2` (`c3t2d0`).

Example – **`format`**(1M) Command for SCSI-Attached Disks

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
          /sbus@1f,0/SUNW,fas@e,8800000/sd@0,0
       1. c0t1d0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
          /sbus@1f,0/SUNW,fas@e,8800000/sd@1,0
       2. c3t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /sbus@1f,0/QLGC,isp@0,10000/sd@0,0
       3. c3t2d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /sbus@1f,0/QLGC,isp@0,10000/sd@2,0
Specify disk (enter its number): *1*
selecting c0t1d0
[disk formatted]
Warning: Current Disk has mounted partitions.

FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        repair     - repair a defective sector
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        inquiry    - show vendor, product and revision
        volname    - set 8-character volume name
        <cmd>      - execute <cmd>, then return
        quit
format> par

PARTITION MENU:
        0      - change "0" partition
        1      - change "1" partition
        2      - change "2" partition
        3      - change "3" partition
        4      - change "4" partition
        5      - change "5" partition
        6      - change "6" partition
        7      - change "7" partition
        select - select a predefined table
        modify - modify a predefined partition table
        name   - name the current table
        print  - display the current table
        label  - write partition map and label to the disk
        <cmd>  - execute <cmd>, then return
        quit
partition> pri
Current partition table (original):
Total disk cylinders available: 2733 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0        var    wm       0 - 2732        1.98GB    (2733/0/0) 4154160
  1 unassigned    wm       0               0         (0/0/0)          0
  2     backup    wm       0 - 2732        1.98GB    (2733/0/0) 4154160
  3 unassigned    wm       0               0         (0/0/0)          0
  4 unassigned    wm       0               0         (0/0/0)          0
  5 unassigned    wm       0               0         (0/0/0)          0
  6 unassigned    wm       0               0         (0/0/0)          0
  7 unassigned    wm       0               0         (0/0/0)          0

partition> q
```

Verifying Disk Space

The software requires a disk cache consisting of redundant arrays of inexpensive disks (RAID) devices, JBOD ("just a bunch of disks") devices, or both. It also requires a certain amount of disk space in the / (root), `/opt`, and `/var` directories. The actual amount needed varies depending on the packages you install.

The following table shows the minimum amount of disk space required in these various directories.

Table – Minimum Disk Space Requirements

Directory	SAM (Archiving and File System)	QFS (File System Only)	SAM-QFS Manager
/ (root) directory	2 Mbytes	2 Mbytes	25 Mbytes
`/opt` directory	21 Mbytes	8 Mbytes	5 Mbytes
`/var` directory	4 Gbytes	1 Mbyte	2 Mbytes
`/usr` directory	2 Mbytes	2 Mbytes	7 Mbytes
`/tmp` directory	0 Mbytes	0 Mbytes	200 Mbytes

> Note -
> The space requirements for the `/var` directory take into account the fact that the archiver data directory, the archiver queue files, and the log files are written to the `/var` directory.

How to Verify Disk Space

The following procedure shows how to verify whether there is enough disk space on your system to accommodate the `SUNWsamfsu` and `SUNWsamfsr` software packages.

1. Run the following command to verify that at least 2 Mbytes are in the `avail` column for the / directory.

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

2. Run the following command to verify that at least 21 Mbytes are in the `avail` column for the `/opt` directory.

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

3. Verify that at least 6 Mbytes are available in the `/var` directory.
A quantity of 30 Mbytes or more is recommended to allow for the growth of log files and other system files.
4. If each directory does not have enough room for the software, repartition the disk to make more space available to each file system.
For information about how to repartition a disk, see your Sun Solaris system administration documentation.

Preparing Hardware for Archiving

Verifying Archive Media

If you plan to perform disk archiving (to archive to disk space in another file system), verify the following:

- The host system to which the disks are attached has at least one file system created that is compatible with the SAM-QFS software.
- The disk has enough space available to accommodate the archive copies.

If you plan to archive to removable media devices, your environment must include the following:

- At least one removable media device for archiving files. This device can be a single tape or optical drive, or it can be multiple devices, such as the drives within an automated library.
- Tape or magneto-optical cartridges to which archive files can be written. For most SCSI-attached and FC-attached libraries, the

SAM-QFS software supports only one media type. If you have a tape library that can be partitioned logically into two or more libraries, you can have one media type in one logical library and a different media type in another. The SAM-QFS software records the cartridges used for each library in a library catalog. You cannot mix the tape media types in a library catalog, so plan to use only one media type per library or logical library.

The SAM-QFS environment supports a wide variety of removable media devices. You can obtain a list of currently supported drives and libraries from your Sun Microsystems sales or support staff.

To make sure that your devices are attached and enumerated in an easily retrieved list, perform one or both of the following procedures:

- If your removable media devices are not attached to your server, perform the procedure in Verifying Archive Media.
- Enumerate your devices using the instructions in Creating a List of Devices. You will use this list again in Installing the Software Packages.

How to Attach Removable Media Devices

The following steps are general guidelines for attaching removable media hardware to a server. For explicit instructions on how to connect these peripherals to a server, refer to the hardware installation guide supplied by the vendor with the automated library and drives.

1. Ensure that you are on a console connection to the server.
2. Power off the server.
3. Ensure that the removable media devices and the disks to be used for the Sun QFS file system are connected and properly addressed.
4. If you have libraries attached to the host system through a SCSI interface, ensure that the SCSI target IDs are unique for each SCSI initiator (host adapter).
   Avoid setting SCSI target IDs for peripherals to IDs that are already in use. In addition, if you are using a SCSI host adapter with a previously attached disk drive, any additional peripheral connected to this bus must have a different ID. Typically, the initiator uses ID 7, and the internal disk drive uses ID 3 for SPARC systems and ID 0 for UltraSPARC systems.
5. Power on the peripherals according to the manufacturer's recommended sequence.
   Typically, you power on the outermost peripherals first, working toward more central components in sequence.
6. Disable autobooting.
   At the `>ok` prompt, type the following command to disable autobooting:

   ```
   >ok setenv auto-boot? false
   ```

7. Type `reset` at the next prompt:

   ```
   >ok reset
   ```

8. Do one of the following:
   - If you have libraries attached to the host system through a SCSI interface, use the `probe-scsi-all` command to conduct an inventory of target IDs and logical unit numbers (LUNs) for each device connected to the host system. Save the output. You will use the information in this output for the next procedure, Creating a List of Devices.
     For example:

     ```
     {0} ok probe-scsi-all
     /pci@6,400/scsi@2,1
     Target 0
       Unit 0   Removable Device type 8     STK 9730        1700
     Target 1
       Unit 0   Removable Tape   type 7     QUANTUM DLT7000 2565
     Target 2
       Unit 0   Removable Tape   type 7     QUANTUM DLT7000 2565
     /pci@1f,4000/scsi@3
     Target 0
       Unit 0   Disk     SEAGATE ST318404LSUN18G 4207
     Target 6
       Unit 0   Removable Read Only device   TOSHIBA XM6201TASUN32XCD1103
     ```

   - If libraries or tape drives are attached to the host system through an FC interface, conduct an inventory of target IDs and LUNs for each device connected to the host system. Save the output. You will use the information in this output for the next procedure, Creating a List of Devices.
     For example:

```
{0} ok show-devs
/SUNW,ffb@1e,0
/SUNW,UltraSPARC-II@2,0
/SUNW,UltraSPARC-II@0,0
/counter-timer@1f,1c00
/pci@1f,2000
/pci@1f,4000
/virtual-memory
/memory@0,a0000000
/aliases
/options
/openprom
/chosen
/packages
/pci@1f,2000/SUNW,qlc@1
/pci@1f,2000/SUNW,qlc@1/fp@0,0
/pci@1f,2000/SUNW,qlc@1/fp@0,0/disk
/pci@1f,4000/SUNW,ifp@2
/pci@1f,4000/scsi@3,1
/pci@1f,4000/scsi@3
/pci@1f,4000/network@1,1
/pci@1f,4000/ebus@1
/pci@1f,4000/SUNW,ifp@2/ses
{0} ok select /pci@1f,2000/SUNW,qlc@1
{0} ok show-children
LiD HA LUN  --- Port WWN ---   ----- Disk description -----
 2  7e   0  500104f00041182b  STK     L700            0236
7c  7e   0  500104f00043abfc  STK     9840            1.28
7d  7e   0  500104f00045eeaf  STK     9840            1.28
6f  7e   0  500104f000416304  IBM     ULT3580-TD1     16E0
6e  7e   0  500104f000416303  IBM     ULT3580-TD1     16E0
```

If the server does not acknowledge all the known devices (disk drives, tape or optical drives, the automated library, and so on), check the cabling. Do not proceed until all devices appear when probed.

9. Reenable autobooting, and then boot the system:

```
>ok setenv auto-boot? true
>ok *boot*
```

10. Review system files.
    Review the following files:
    - `/var/adm/messages` to ensure that all devices were recognized
    - `/dev/rmt` for expected tape devices
    - `/dev/dsk` and `/dev/rdsk` for expected disks
      Due to special driver requirements, no device information appears in `/var/adm/messages` for magneto-optical devices or libraries until after you install the SAM-QFS software packages.
11. Disable autocleaning and autoloading.
    If your automated library supports autocleaning or autoloading, disable those features when using that library with the Sun Storage Archive Manager software. Consult the documentation from your library's manufacturer for information about disabling autocleaning and autoloading.

> Note -
> The only times you can use autoloading are during the initial loading of cartridges and when the SAM-QFS software is not running. Remember to disable autoloading when the SAM-QFS system is running.

Creating a List of Devices

The devices that you intend to use must be attached and recognized by the server upon which you intend to install the SAM-QFS software. To configure the SAM-QFS software, you need to know the following about your devices:

- The device type, manufacturer, and model number.
- The mechanism by which the device is attached to the server. You can attach devices in one of the following ways:

- Drives can use either a SCSI attachment or an FC attachment. Each drive accepts either tape cartridges or magneto-optical cartridges.
  For SCSI-attached drives, you need to know each drive's SCSI target ID and logical unit number (LUN).
  For FC-attached drives, you need to know each drive's LUN and node World Wide Name (WWN).
- Automated libraries can use a SCSI attachment, an FC attachment, or a network attachment.
  Libraries that use SCSI or FC attachments are called direct attached libraries. For SCSI-attached libraries, you need to know each library's SCSI target ID and LUN. For FC-attached libraries, you need to know each library's LUN and node WWN.
  Libraries that use a network attachment are called network attached libraries. You cannot configure network attached libraries in the existing system configuration files; instead, you must create a parameters file for each network attached library. This is explained later in the installation process.

How to Create a List of Devices

Fill in the following table to include the name, manufacturer, model, and connection types for each device that you want to include in your SAM-QFS environment. Retain this list for use again later in the configuration procedure.

Device Name, Manufacturer, and Model	Target ID	LUN	Node WWN
SCSI-attached tape drives			
			Not applicable
			Not applicable
			Not applicable
FC-attached tape drives			
	Not applicable		
	Not applicable		
	Not applicable		
SCSI-attached magneto-optical drives			
			Not applicable
			Not applicable
			Not applicable
FC-attached magneto-optical drives			
	Not applicable		
	Not applicable		
	Not applicable		
SCSI-attached automated libraries			
			Not applicable
			Not applicable
			Not applicable
FC-attached automated libraries			
	Not applicable		
	Not applicable		
	Not applicable		

Obtaining the Release Files

Make sure that you have a copy of the release software. You can obtain the Sun SAM-QFS software from the Sun Download Center or on a

CD-ROM. Contact your authorized service provider (ASP) or your Sun sales representative if you have questions on obtaining the software.

After the release, upgrade patches are available from Sun Solve.

> Caution -
> If you have not read the Release Notes, do so before you continue.

How to Obtain the Software From the Sun Download Center

1. Go to the downloads page.
2. Select the SAM-QFS or Sun QFS software package you want to receive.
3. Follow the instructions on the web site for downloading the software.

Software Licensing

You must agree to all binary and right-to-use (RTU) software license agreements before you install either the Sun QFS or the Sun SAM software.

Setting Up the Network Management Station

Perform this procedure if you want to monitor your configuration through Simple Network Management Protocol (SNMP) software.

You can configure the Sun SAM-QFS software to notify you when potential problems occur in its environment. The SNMP software manages information exchange between network devices such as servers, automated libraries, and drives. When the Sun SAM-QFS software detects potential problems in its environment, it sends information to a management station, which enables you to monitor the system remotely.

The management stations you can use include the following:

- Sun Storage Automated Diagnostic Environment (StorADE)
- Sun Management Center (Sun MC)
- Sun Remote Server (SRS)
- Sun Remote Services Net Connect

If you want to enable SNMP traps, make sure that the management station software is installed and operating correctly before installing the Sun SAM-QFS software. Refer to the documentation that came with your management station software.

The types of problems, or events, that the SAM-QFS software can detect are defined in the SAM-QFS Management Information Base (MIB). The events include errors in configuration, `tapealert`(1M) events, and other atypical system activity. For complete information about the MIB, see `/var/snmp/mib/SUN-SAM-MIB.mib` after the packages are installed.

The SAM-QFS software supports the TRAP SNMP (V2c) protocol. The software does not support `GET-REQUEST`, `GETNEXT-REQUEST`, and `SET_REQUEST`.

Release Package Contents, Directories, and Files

Release Package Contents

The Sun Storage Archive Manager (SAM-QFS) and Sun QFS software packages are in Sun Solaris `pkgadd`(1M) format. These packages reflect the Sun Solaris version for the platform on which you will be installing the software.

The following table shows the release package names.

Table – Release Package Names

Installed Package	Description
`SUNWqfsr` `SUNWqfsu`	Sun QFS (file system only) software packages

SUNWsamfsr SUNWsamfsu	SAM-QFS (archiving and file system) software packages
SUNWfsmgrr SUNWfsmgru	SAM-QFS Manager software packages; Run the `fsmgr_setup` script to install
SUNWsamfswm	WORM-FS support packages

The releases are identified using characters arranged in the following format:
major U update . patch

The U stands for "update" in this format.

In the patch number field, a number between 1 and 99 indicates a patch release, and a letter from A through Z indicates pre-release software. The base release of a first feature release of a major release might not contain a patch level.

For example:

- 4U0 is release 4, update 0, a major release with no minor release revisions and no bug fixes.
- 4U2 is release 4, update 2, a minor release.
- 4U2.1 is a patch release that contains software fixes for a major or minor release. This number appears in the patch's README file.

Directories and Files Created

This section describes the directories and files associated with the Sun QFS and SAM-QFS products. You can obtain additional information about the files in this section from the man pages after the software is installed.

Directories Created at Installation

The following table lists the directories created when the software packages are installed.

Table – Directories Created

Directory	Content
`/dev/samst`	Device driver special files (only when SAM-QFS packages are installed).
`/etc/fs/samfs`	Commands specific to the software.
`/etc/opt/SUNWsamfs`	Configuration files.
`/etc/opt/SUNWsamfs/scripts`	Site-customizable scripts.
`/opt/SUNWsamfs/bin`	User command binaries.
`/opt/SUNWsamfs/client`	Files for remote procedure call API client.
`/opt/SUNWsamfs/doc`	Documentation repository for any informational files included in the release. The README file, which summarizes the installed release's features, is included in this directory.
`/opt/SUNWsamfs/examples`	Various example configuration files.
`/opt/SUNWsamfs/include`	API include files.
`/opt/SUNWsamfs/lib`	Relocatable libraries.
`/opt/SUNWsamfs/man`	Man pages.
`/var/snmp/mib`	Standard MIB files and product MIB (`SUN-SAM-MIB.mib`).
`/opt/SUNWsamfs/sbin`	System administrator commands and daemon binaries.
`/opt/SUNWsamfs/sc`	Sun Cluster binaries and configuration files.
`/opt/SUNWfsmgr/bin`	SAM-QFS Manager administrator commands.
`/opt/SUNWfsmgr/doc`	SAM-QFS Manager online documentation repository.
`/var/opt/SUNWsamfs`	Device catalogs, catalog trace file, log files, and archiver data directory and queue files.

Files Created at Installation

The following table lists miscellaneous files created when the software is installed.

Table – Files Created - Miscellaneous

File	Description
`/etc/opt/SUNWsamfs/inquiry.conf`	Vendor and product identification strings for recognized SCSI devices (only when SAM-QFS packages are installed).
`/etc/sysevent/config/SUNW,SUNWsamfs,sysevent.conf`	Solaris system event handler configuration file.
`/kernel/drv/amd64/samaio`	File system asynchronous I/O pseudo-driver (64-bit version for x64 platforms).
`/kernel/drv/amd64/samioc`	Sun Solaris 64-bit file system interface module (for x64 platforms).
`/kernel/drv/amd64/samst`	SAM-QFS driver for SCSI media changers and optical drives for tape drives (64-bit version for x64 platforms).
`/kernel/drv/samaio.conf`	Configuration file for `samaio`.
`/kernel/drv/samioc.conf`	Configuration file for the `samioc` module.
`/kernel/drv/samst.conf`	Configuration file for the `samst` driver.
`/kernel/drv/sparcv9/samaio`	File system asynchronous I/O pseudo-driver (64-bit version for SPARC platforms).
`/kernel/drv/sparcv9/samioc`	Sun Solaris 64-bit file system interface module (for SPARC platforms).
`/kernel/drv/sparcv9/samst`	SAM-QFS driver for SCSI media changers and optical drives for tape drives (64-bit version for SPARC platforms).
`/kernel/fs/amd64/samfs`	Sun Solaris 64-bit file system module for the x64 platform.
`/kernel/fs/sparcv9/samfs`	Sun Solaris 64-bit file system module for SPARC platforms.
`/var/log/webconsole/host.conf`	SAM-QFS Manager configuration file.
`/var/opt/SUNWsamfs/faults`	Faults history file.
`/var/sadm/samqfsui/fsmgr_uninstall`	Software for removing SAM-QFS Manager and its supporting applications.
`/opt/SUNWsamfs/sc/etc/SUNW.qfs`	Sun Cluster configuration file created only in the presence of Sun Cluster software.
`/usr/cluster/lib/rgm/rtreg/SUNW.qfs`	Sun Cluster configuration file created only in the presence of Sun Cluster software.

The file system has dynamically loadable components that are stored in the Sun Solaris `/kernel` directory. You can use `modinfo`(1M) command to determine the modules that are loaded. Typically, the kernel loads the file system module at boot time. Alternatively, you can load the file system module when the file system is first mounted after the Sun software is installed.

Fault Notification Files

After the software is installed, it creates files that it uses for fault notification. The following table lists these files. When the software detects faults serious enough to merit user attention, the software uses these trap and log files to convey fault information through the SAM-QFS Manager software.

Table – Files Created - Fault Notification

File	Description
`/etc/opt/SUNWsamfs/scripts/sendtrap`	Sends trap information.

/opt/SUNWsamfs/sbin/fault_log	Records faults.
/opt/SUNWsamfs/sbin/tapealert_log	Records tapealert(1M) faults (only when SAM-QFS packages are installed).
/opt/SUNWsamfs/sbin/tapealert_trap	Sends tapealert(1M) traps (only when SAM-QFS packages are installed).

The software creates these files with the following permissions:

```
-rwxr-x---
```

Caution -
Do not change these file permissions.
If execute permissions are lost, for example, the system writes messages such as the following to /var/adm/messages:

```
SUNW,SUNWsamfs,sysevent.conf, line1: no execute access to
/opt/SUNWsamfs/sbin/tapealert_trap - No such file or directory.
```

Site Files

The configuration procedures elsewhere in this information direct you to create several site-specific files. The software uses these site files.

Note -
Your site's configuration files must contain ASCII characters only.

You must create the master configuration mcf file at your site in order to use the Sun SAM-QFS software. For more information about the /etc/opt/SUNWsamfs/mcf file, see About the Master Configuration File and the mcf(4) man page.

If you are using the archiver and file system features, you might also create all the files shown in the following table. If you are using only file system features, you might only create the first two files.

Table – Optional Site Files

File	Description
/etc/opt/SUNWsamfs/samfs.cmd	File system mount parameter command file. For more information, see the samfs.cmd(4) man page.
/etc/opt/SUNWsamfs/defaults.conf	Miscellaneous default values. For more information, see the defaults.conf(4) man page.
/etc/opt/SUNWsamfs/archiver.cmd	Archiver command file. For more information, see the archiver.cmd(4) man page, or Configuring the Archiver.
/etc/opt/SUNWsamfs/preview.cmd	Previewer command file. For more information, see Configuring the Stager and the preview.cmd(4) man page.
/etc/opt/SUNWsamfs/recycler.cmd	Recycler command file. For more information, see Configuring the Recycler and the recycler.cmd(4) man page.
/etc/opt/SUNWsamfs/releaser.cmd	Releaser command file. For more information, see About Releasing and the releaser.cmd(4) man page.

Modified System Files

During installation, the software adds information to certain Solaris system files. These system files are ASCII text files. The Solaris OS uses these files to identify loadable kernel modules by number rather than by name.

The software adds information to the following files:

- `/etc/name_to_major`
  The SAM-QFS software uses this file to map drivers to major numbers. The `samst` and `samrd` major numbers can vary, depending on the major numbers in use by the Solaris OS. The system adds the following lines to this file:

```
samst 63
samrd 64
samioc 236
samaio 237
```

- `/etc/security/auth_attr`
  This file is the authorization description database. The system adds the following lines to this file:

```
# File System Manager Authorizations
com.sun.netstorage.fsmgr.config:::File System Manager All Access::
com.sun.netstorage.fsmgr.operator.media:::File System Manager Media
Related Operation Access::
com.sun.netstorage.fsmgr.operator.sam.control:::File System Manager
Start/Stop/Idle Archiving Access::
com.sun.netstorage.fsmgr.operator.file:::File System Manager File
LevelOperation Access::
com.sun.netstorage.fsmgr.operator.filesystem:::File System Manager
FileSystem Level Operation Access::
```

- `/etc/user_attr`
  This file is the extended user attributes database used by SAM-QFS Manager.

```
root::::profiles=Web Console Management,All;auths=
Solaris.*,solaris.grant,*com.sun.netstorage.fsmgr.**;
lock_after_retries=no
```

- `/etc/inittab`
  The system adds the following line to this file:

```
sfad:3:respawn:/opt/SUNWsamfs/sbin/fsmgmtd
```

Installing and Configuring SAM-QFS

The Sun QFS and Sun Storage Archive Manager (SAM-QFS) products are closely linked. Depending on the features that you need, choose from the following:

Feature	Packages	Required License	For More Information
Archiving (local file system)*	`SUNWsamfsr`, `SUNWsamfsu`	SAM-QFS	Read this document.
Archiving (shared file system)	`SUNWsamfsr`, `SUNWsamfsu`	SAM-QFS and QFS	Read this document.
Local or shared file system (no archiving)	`SUNWqfsr`, `SUNWqfsu`	QFS	See Installing Sun QFS.
Browser-based management	`SUNWfsmgrr`, `SUNWfsmgru`	SAM-QFS or QFS	Use `fsmgr_setup` as explained in Installing SAM-QFS Manager.

* Formerly referred to as SAM-FS.

Complete the tasks below if this is the initial installation of the SAM-QFS (archiving and file system) software packages at your site.

> Note
> You must be logged in as superuser to complete the installation tasks.

Before You Begin

- If you are not already familiar with Sun QFS and SAM-QFS, see About QFS and SAM-QFS.
- Before you follow the detailed installation steps below, check the hardware and software requirements as explained in Preparing for Installation.

Installation Overview Task Map

Depending on the the features that you need to support, you must complete several of the following procedures.

Step	Task	Description
1	Add the software packages.	Install the appropriate packages for your needs. Also see Release Package Contents.
2	Configure `path` and `manpath` variables.	Configure the environment variables for access to commands and man pages.
3	(Optional) Install and configure SAM-QFS Manager.	This task is needed only if you want to use a browser to configure file systems and archiving features.
4	(Optional) Configure tape and magneto-optical storage devices.	This task is needed only if you want to use tape or magneto-optical devices for archiving.
5	Configure the file system environment.	Define the master configuration file `mcf`.
6	Configure file system mount parameters.	Define the `/etc/vfstab` and `samfs.cmd` files.
7	Initialize the environment.	Initialize SAM-QFS and mount the file systems.
8	Configure file archiving.	Define parameters for archiving file systems to media.

Installing the Software Packages

The Sun QFS and Sun Storage Archive Manager (SAM-QFS) software uses the Sun Solaris packaging utilities for adding and deleting software. The `pkgadd`(1M) utility prompts you to confirm various actions necessary to install the packages.

Table – Release Package Names

Installed Package	Description
`SUNWqfsr` `SUNWqfsu`	Sun QFS (file system only) software packages
`SUNWsamfsr` `SUNWsamfsu`	SAM-QFS(archiving and file system) software packages

How to Add the Packages

1. Become superuser.
2. Use the `cd` command to go to the directory where the software package release files reside.
   You obtained the release files as described in Obtaining the Release Files. Changing to the appropriate directory differs, depending on your release media, as follows:
   - If you downloaded the release files, go to the directory to which you downloaded the files.
   - If you obtained the release files from a CD-ROM, go to the directory on the CD-ROM that corresponds to your operating system version.
3. Use the `pkgadd`(1M) command to add the appropriate packages, based on the features that you need to support:

- For archiving to a local or shared file system, install the SUNWsamfsr and SUNWsamfsu packages.
- To support just a local or shared file system (no archiving), install the SUNWqfsr and SUNWqfsu packages.
  For example:

```
# pkgadd -d . SUNWsamfsr SUNWsamfsu
```

4. When prompted to define an administrator group, select yes or y to accept the default (no administrator group), or select no or n if you want to define an administrator group.
   You can reset permissions on certain commands later by using the set_admin(1M) command. For more information about this command, see Adding the Administrator Group or the set_admin(1M) man page.
5. Examine the SAM-QFS installation log file /tmp/SAM_install.log.
   This file should show that the pkgadd(1M) command added the SUNWsamfsr and SUNWsamfsu software packages. Verify that the SAM-QFS samst driver is also installed. If all files installed properly, the following message appears:

```
Restarting the sysevent daemon
```

How to Set Up PATH and MANPATH Variables

To enable access to the commands and man pages for the Sun QFS and SAM-QFS commands, you must modify your PATH and MANPATH environment variables.

1. For users who will need to access the user commands, such as sls(1), add /opt/SUNWsamfs/bin to the users' PATH variables.
2. Edit your system setup files to include the correct paths to commands and man pages.
   - In the Bourne or Korn shells, change the PATH and MANPATH variables in the .profile file and export the variables.
     The following example shows how your .profile file might look after editing.

```
PATH=$PATH:/opt/SUNWsamfs/bin:/opt/SUNWsamfs/sbin
MANPATH=$MANPATH:/opt/SUNWsamfs/man
export PATH MANPATH
```

   - In the C shell, edit the .login and .cshrc files.
     The path statement in your .cshrc file might look like the following example:

```
set path = ($path /opt/SUNWsamfs/bin /opt/SUNWsamfs/sbin)
```

     The MANPATH statement in your .login file might look like the following example:

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

Installing and Configuring SAM-QFS Manager

If you want to use the browser user interface to configure and manage your SAM-QFS environment, follow the instructions in this section.

About SAM-QFS Manager

The SAM-QFS Manager is a browser interface tool that enables you to configure, control, protect, and monitor the archiving and file systems in your network from a central location. To access this central location, you can use the web browser on any host in your network.

The goal of the software is to provide a less complex way than command-line interface (CLI) commands to perform the most common archiving and file system tasks.

By default, SAM-QFS Manager is set up to manage the server on which it is installed. It can also be used to manage other servers running Sun Storage Archive Manager software, but those additional servers must first be configured to allow SAM-QFS Manager access. For instructions on how to add additional managed servers, see How to Add an Additional Server for SAM-QFS Manager Access.

Installing SAM-QFS Manager

Perform the tasks in this section to install the SAM-QFS Manager software.

Verifying Requirements for SAM-QFS Manager

You can install the SAM-QFS Manager software in one of the following configurations:

- As a stand-alone management station to manage one or more SAM-QFS Manager hosts
- As additional software on the SAM-QFS Manager host

After the SAM-QFS Manager software is installed, you can invoke SAM-QFS Manager from any machine on the network that is allowed access to its web server.

The host upon which you are configuring SAM-QFS Manager must meet the requirements described in the following sections.

Hardware Requirements

The minimum hardware requirements for the SAM-QFS Manager software are as follows:

- SPARC 400-MHz (or more) CPU or x64 AMD CPU
- 1 gigabyte of memory
- One 20-gigabyte disk
- At least 250 Mbytes free space in `/tmp`
- At least 100 Mbytes free space in / (the root partition)
- One 10/100/1000Base-T Ethernet port

Browser Requirements

Ensure that your installation meets the following browser requirements:

- One of the following browsers, at the minimum level indicated, must be used to access the File System Manager software:
  - Netscape™ 7.1 / Mozilla™ 1.7 / Firefox 1.5 on the Solaris OS or Microsoft Windows 98 SE, ME, 2000, or XP operating system
  - Internet Explorer 6.0 on the Microsoft Windows 98 SE, ME, 2000, or XP operating system
- You must enable JavaScript™ technology in your browser. Specific locations differ by browser. For example, in Firefox, go to the Content section of the Preferences panel and confirm that the box next to Enable JavaScript is checked.
- If you are upgrading from an earlier version of the SAM-QFS Manager software, you must clear the browser cache before using SAM-QFS Manager for the first time.

Operating System Requirements

Make sure that the following minimum level of the Solaris OS is installed on the web server:

- Solaris 10 Update 6

Web Software Requirements

The SAM-QFS Manager installation packages include revisions of the following software at the minimum levels indicated:

- Java™ 2 Standard Edition version 1.5.0
- JavaHelp™ 2.0
- Java Studio Enterprise Web Application Framework (JATO) 2.1.2
- Tomcat version 4.0.5

During the installation procedure, you will be asked to answer questions. Based on your answers, the installation software can install the correct revisions for you if the compatible revisions of these software packages are not present.

> Note -
> SAM-QFS Manager is registered in the Sun Java Web Console and can coexist with other applications that use the same console. The Java Web Console uses port 6789. This is an IANA-reserved port, so no application other than Java Web Console should use this port.

How to Install SAM-QFS Manager

Before You Begin

Ensure that you have met the installation requirements described in Verifying Requirements for SAM-QFS Manager.

Steps

1. Log in to the server that you want to use as the SAM-QFS management station.
   You can use the same server on which you installed the `SUNWsamfsr` and `SUNWsamfsu` packages, or you can use a different server on the same network.
2. Become superuser.
3. Go to the directory where the software package release files reside on your server.
4. Execute the `fsmgr_setup` script to start the installation process.
   For example:

```
# fsmgr_setup
```

5. Answer the questions as prompted by the `fsmgr_setup` script.
   During the installation procedure, you are asked questions about your environment.
   The `fsmgr_setup` script automatically installs the following:
   - The `SUNWfsmgrr` package.
   - The `SUNWfsmgru` package.
     The installation script prompts you to specify whether you want to install localized packages.
     After installing the packages, the install software starts the Tomcat Web Server and enables logging.
6. Edit your system setup files to include the correct paths to commands and man pages.
   - In the Bourne or Korn shell, edit the `.profile` file, change the `PATH` and `MANPATH` variables, and export the variables.
     The following code example shows how your `.profile` file might look after editing.

```
PATH=$PATH:/opt/SUNWfsmgr/bin
MANPATH=$MANPATH:/opt/SUNWfsmgr/man
export PATH MANPATH
```

   - In the C shell, edit the `.login` and `.cshrc` files.
     When you have finished editing, the `path` statement in your `.cshrc` file might look like the following line:

```
set path = ($path /opt/SUNWfsmgr/bin)
```

     The following code example shows how the `MANPATH` in your `.login` file might after you have finished editing.

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

7. Use the `ps`(1) and `grep`(1) commands to make sure that the `rpcbind` service is running:

```
# ps -ef | grep rpcbind
```

8. Examine the output from the preceding commands.
   The output should contain a line similar to the following:

```
root    269     1  0    Feb 08 ?        0:06 /usr/sbin/rpcbind
```

   If `rpcbind` does not appear in the output, type the following command to start the `rpcbind` service:

```
# /usr/sbin/rpcbind
```

9. (Optional) Start the SAM-QFS Manager (`fsmgmtd`) daemon.
   If you did not choose to start the SAM-QFS Manager daemon automatically during the installation process, do one of the following:
   - Type the following command to start the SAM-QFS Manager daemon and have it restart automatically every time the daemon process dies. With this configuration, the daemon also automatically restarts at system reboot.

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

   - Type the following command if you want the SAM-QFS Manager daemon to run only once and not automatically restart.

     ```
     # /opt/SUNWsamfs/sbin/fsmadm start
     ```

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

10. (Optional) Give additional users access to SAM-QFS Manager.
    By default, the `root` user has privileges to perform all operations available from the SAM-QFS software. You can assign other users full or partial access to SAM-QFS Manager operations.
    To give an additional user access to SAM-QFS Manager, use the `useradd` command. See How to Create Additional SAM-QFS User Accounts and How to Assign Privilege Levels to SAM-QFS Users for information about adding users and assigning SAM-QFS Manager user privilege levels.

How to Start SAM-QFS Manager

> Note -
> Before you start SAM-QFS Manager, disable all pop-up blockers.

1. Log in to the server where SAM-QFS Manager is installed, or log in to any system that has network access to it.
2. If you upgraded from a previous version of the software, open the web browser and clear the browser cache.
3. From the web browser, start SAM-QFS Manager.

   ```
   https://<hostname>:6789
   ```

   For hostname, type the name of the host where the SAM-QFS Manager software is installed. If you need to specify a domain name in addition to the host name, specify the hostname in this format: hostname.domainname. Note that this URL begins with `https`, not `http`.
   The Sun Java Web Console login page is displayed.

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

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

5. At the Password prompt, type the password.
6. Click Log In.
7. In the Storage section of the Applications page, select SAM-QFS Manager.
   You are now logged in to SAM-QFS Manager.

Configuring SAM-QFS Manager

After SAM-QFS Manager is installed, you can log in to the software using the `root` user name and the password for the management station.

The root login grants you full administrator privileges to configure, monitor, control, and reconfigure the devices in your SAM-QFS Manager environment. Only the SAM-QFS Manager administrator should log in using the root login. All other users should log in using another user name.

By default, SAM-QFS Manager is set up to manage the server on which it is installed. It can also be used to manage other servers running SAM-QFS Manager software, but those additional servers must first be configured to allow SAM-QFS Manager access. For instructions on adding additional managed servers, see How to Add an Additional Server for SAM-QFS Manager Access.

How to Create Additional SAM-QFS User Accounts

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

If you remove the SAM-QFS Manager software, the removal scripts do not remove any additional accounts that you create manually. You must use one or both of the following procedures to administer any accounts you add manually.

Steps

1. Outside of the browser interface, log in to the SAM-QFS management station as `root`.
2. Use the `useradd` and `passwd` commands to add each user.
   For example, to add a user with account name `bobsmith`, type the following:

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

   Each user account that you add in this way has read-only viewing privileges for SAM-QFS Manager functions. To add additional privileges see How to Assign Privilege Levels to SAM-QFS Users.

How to Assign Privilege Levels to SAM-QFS Users

You can assign users full or partial access to SAM-QFS Manager functions.

1. Log in to the SAM-QFS management station as root.
2. To specify full or partial configuration privileges for a user, add the following line to the `/etc/user_attr` file.
   account-name**::::auths=**privilege-level
   - account-name is the name of the user's account.
   - privilege-level is the level of authorization to assign to the user.
     The following table lists the five levels of privileges you can assign to SAM-QFS Manager users.

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

Example – Assigning Full Privileges to a User

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

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

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

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

How to Create a SAM-QFS Manager Account for Multiple Users

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

1. Use the useradd and passwd commands to add the account.
   For example, to add a user account called guest for multiple users, type the following:

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

2. Use the roleadd and passwd commands to add the role.
   To create a role called admin with special privileges within the guest account, type the following:

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

3. Specify the privilege levels in the /etc/user_attr file.
   To assign the admin role privileges to restore and stage file systems, add the following lines to the /etc/user_attr file:

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

   In this example, when a user logs in as guest, SAM-QFS Manager prompts the user to select either No Role or Admin. If users know the Admin role password, they can select Admin, provide the Admin password, and have privileges to restore and stage file systems. All other users must select No Role and have read-only privileges.
   Because multiple users with the same privilege level can be logged in to the software concurrently, one user's changes could potentially overwrite another user's previous changes. To prevent this situation, develop policies about who can make changes and how to notify others.

How to Add an Additional Server for SAM-QFS Manager Access

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

1. Outside of the browser interface, use the telnet utility to connect to the server you want to add.
   Log in as root.
2. Use the fsmadm(1M) add command to add the SAM-QFS management station to the list of hosts that can remotely administer this server.
   Only hosts that are added to the list through this command can remotely administer the server.
   For example:

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

3. To ensure that the SAM-QFS management station is successfully added, use the fsmadm(1M) list command and verify that your SAM-QFS management station is listed in the output.
4. Log in to the SAM-QFS Manager browser interface as an administrator user.
5. On the Servers page, click Add.
   The Add Server window is displayed.

6. In the Server Name or IP Address field, type the name or the IP address of the new server.
7. Click OK.

How to Set the SAM-QFS Manager Session Timeout

The Java Web Console framework has a default session timeout of 15 minutes. The SAM-QFS Manager installation program changes the session timeout to 60 minutes. You can change the session timeout to a different value, but to preserve security you should not set it to a value greater than 60 minutes.

1. To change the session timeout value, enter the following command on the SAM-QFS management station:

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

Example – Changing the Session Timeout Value

To change the timeout value to 45 minutes, you would type:

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

Configuring Tape and Magneto-Optical Storage Devices

Configuring Storage Devices for Archiving

Perform the tasks in this section if you plan to enable archiving to tape or magneto-optical media. If you plan to archive to disk, you do not need to perform these tasks.

This section describes how to verify and update the following files:

- `/kernel/drv/st.conf`, which configures tape drives attached to the server through a SCSI or FC attachment.
- `/kernel/drv/samst.conf`, which configures the following devices that the Sun Storage Archive Manager (SAM) software recognizes by default:
  - Direct attached automated libraries
  - Magneto-optical drives attached to the server through a SCSI or FC attachment

The SAM package includes the `/opt/SUNWsamfs/examples/st.conf_changes` file, which contains configuration information for tape drives that are not supported in the Solaris kernel by default.

Task Map: Configuring Storage Devices for Archiving

Depending on the devices you want to configure, you must complete one or more of the following procedures.

Step	Task	Description
1	Create List of Devices	Take an inventory of your devices to configure.
2	Add Tape Devices	Perform this task for each tape drive that you want to add to the SAM environment.
3	Add Tape Drive Interface Target IDs and LUNs	Perform this task if your tape drives are attached through a SCSI or FC interface.
4	Add Libraries or Magneto-Optical Drives	Perform this task if you have any magneto-optical drives, SCSI-attached automated libraries, or FC-attached automated libraries that you want to include in your SAM environment.
5	Verify Configured Devices	Verify that you have all of your devices configured properly.
6	Enable the Storage Device Configuration	Reboot the system to enable the changes you have made.

7	Create Parameters Files for Network Attached Automated Libraries	Create parameter files for the network attached automated libraries.

The procedures in this task include an example that is based on the inventory list shown in the following table.

Example Inventory List - Devices to Be Configured

Device Name, Manufacturer, and Model	Target ID	LUN	Node WWN
SCSI-attached tape drives			
QUANTUM DLT7000	1	0	Not applicable
QUANTUM DLT7000	2	0	Not applicable
FC-attached tape drives			
StorageTek 9840	Not applicable	0	500104f00043abfc
StorageTek 9840	Not applicable	0	500104f00045eeaf
IBM ULT3580-TD1	Not applicable	0	500104f000416304
IBM ULT3580-TD1	Not applicable	0	500104f000416303
SCSI-attached automated libraries			
StorageTek 9730	0	0	Not applicable
FC-attached automated libraries			
StorageTek L700	Not applicable	0	500104f00041182b

Note -
The device names are shown as they appear in the discovery output.

How To Add Tape Devices for Archiving

Perform this procedure to include your tape drives in your archiving environment. In this procedure, you make an entry in the `st.conf` file for each unique tape drive that is on your inventory list.

Tip -
You can also add tape devices from SAM-QFS Manager. For information, see How to Add Tape Devices From SAM-QFS Manager.

1. Copy `/kernel/drv/st.conf` to a backup file. For example:

```
# cp /kernel/drv/st.conf /kernel/drv/st.conf.orig
```

2. Open the `/kernel/drv/st.conf` file and find the following string:

```
#tape-config-list=
```

3. Remove the pound character (#).
4. Open the `/opt/SUNWsamfs/examples/st.conf_changes` file. For each tape drive on your inventory list, do the following.
   a. Search for the device definition entry.
      For example, searching for the Quantum DLT 7000 tape drive that is in the example inventory locates the following entry:

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

b. Copy the entry from the `st.conf_changes` file to the `st.conf` file, placing it after the `tape-config-list` line. The following example shows the resulting `st.conf` file.

```
...
tape-config-list=
"QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape",
...
```

c. Note the final string in the entry, which is enclosed in quotation marks. In this example, the final string is "`dlt7-tape`". This string is the tape configuration value.

d. Search `/opt/SUNWsamfs/examples/st.conf_changes` to find the line that begins with this tape configuration value. In this example, the value is:

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

e. Copy the tape configuration value to the `st.conf` file, placing it after the device definition line. The following example shows the lines now contained in the `st.conf` file.

```
...
tape-config-list=
"QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape";
dlt7-tape =    1,0x36,0,0xd679,4,0x82,0x83,0x84,0x85,3;
...
```

f. Repeat this procedure for each type of tape drive.

> Note -
> In the `st.conf_changes` file, a tape configuration value is repeated for each device definition that uses the same tape configuration. In the `st.conf` file, include only one entry for each tape configuration value. For example, the Sony SDT-5000 and the Sony SDT-5200 tape drives both use "DAT" as the final string. A single entry for the DAT tape configuration value is sufficient.

5. Replace the comma (`,`) at the end of the last device definition line with a semicolon (`;`).
   The following example shows a `st.conf` file with definitions for the Quantum DLT 7000, the StorageTek 9840, and the IBM ULT3580 tape drives. The semicolon is after "`CLASS_3580`"

```
...
tape-config-list=
"QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape",
"STK     9840", "STK 9840 Fast Access", "CLASS_9840",
"IBM     ULT3580-TD1",  "IBM 3580 Ultrium",     "CLASS_3580";
dlt7-tape =    1,0x36,0,0xd679,4,0x82,0x83,0x84,0x85,3;
CLASS_9840 = 1,0x36,0,0x1d679,1,0x00,0;
CLASS_3580 = 1,0x24,0,0x418679,2,0x00,0x01,0;
...
```

6. Save your changes.
7. Close the `/opt/SUNWsamfs/examples/st.conf_changes` file.
   Do not close the `st.conf` file because you continue to edit the file in the next procedure.
8. Go to the #How To Add Tape Drive Interface Target IDs and LUNs for Archiving procedure.

How To Add Tape Drive Interface Target IDs and LUNs for Archiving

For each tape drive on your hardware inventory list that is attached through a SCSI or FC interface, you must confirm that an entry in the `st.conf` file defines that interface. This procedure shows how to verify and, if necessary, add target ID and LUN entries.

> **Note** -
> Do not use this procedure to add interface information for magneto-optical drives. See
> How To Add Libraries or Magneto-Optical Drives for Archiving

1. Open the file `/kernel/drv/st.conf` if it is not already open.
2. To use the SCSI interface to attach tape drives, do the following:
    a. Search for entries that have the following format to locate the list of SCSI target IDs and LUNs:

    ```
    name="st" class="scsi" target=<target> lun=<lun>;
    ```

    target is the target ID for each SCSI drive found.
    lun is the corresponding LUN for each SCSI drive found.

    b. Within the list, find each entry that corresponds to each SCSI target and LUN on your hardware inventory list. The following example shows the two entries that correspond to the two Quantum DLT 7000 drives that are attached to LUN 0 and have target IDs 1 and 2, shown in Example Inventory List.

    ```
    name="st" class="scsi" target=1 lun=0;
    name="st" class="scsi" target=2 lun=0;
    ```

    Note that an entry might extend over two lines.

    - If the entry is preceded by a pound character (#), delete the character. A pound character marks a line as a comment.
    - If the entry is missing, create an entry for the SCSI target and LUN line you need, following the format shown in Step a and using the information in your hardware inventory list.
3. To use the FC interface to add tape drives and if you are not using the Sun StorageTek SAN Foundation Software I/O stack, create a line for each FC-attached device after the SCSI target ID and LUN list, using the following format:

    ```
    name="st" parent="fp" lun=<lun> fc-port-wwn="<world-wide-name>"
    ```

    For lun, specify the LUN for the drive.
    For world-wide-name, specify the World Wide Name (WWN) for the drive.
    The following example shows the lines that support the StorageTek 9840 and IBM ULT3580 tape drives included in the Example Inventory List.

    ```
    name="st" parent="fp" lun=0 fc-port-wwn="500104f00043abfc"
    name="st" parent="fp" lun=0 fc-port-wwn="500104f00045eeaf"
    name="st" parent="fp" lun=0 fc-port-wwn="500104f000416304"
    name="st" parent="fp" lun=0 fc-port-wwn="500104f000416303"
    ```
4. Save your changes and close the `st.conf` file.
5. To include magneto-optical drives or automated libraries attached through a SCSI or a Fibre Channel interface, go to the How To Add Libraries or Magneto-Optical Drives for Archiving procedure.

How to Add Tape Devices From SAM-QFS Manager

To add a tape device from the SAM-QFS Manager:

1. In the left pane, click Archive Media.
2. In the Tape Library Summary window, click Add.
3. Follow the steps in the wizard.

How To Add Libraries or Magneto-Optical Drives for Archiving

The `inquiry.conf` file lists all devices that are supported.
The `/kernel/drv/samst.conf` file contains a list of SCSI and FC entries and works with the

`/opt/SUNWsamfs/examples/inquiry.conf` file to define the devices that can be included in a SAM environment.
The following procedures show how to verify the entries in the `samst.conf` and to update the file if necessary.

You update the `samst.conf` file, depending on your environment:

- If you have only network-attached automated libraries, you do not need to verify device support or update it.
- If you use a SCSI or FC interface to attach a tape library to the server, use the procedure in How To Configure Device Support in SCSI or FC Environments to discover the tape libraries with the current drive information.
- If you have a direct attached library with a target number larger than 6 or a LUN identifier larger than 1, use the procedure in How To Configure Device Support for a Direct Attached Library.

How To Configure Device Support in SCSI or FC Environments

Repeat this procedure for each device in your environment.

1. On the SAM-QFS Manager Managed Hosts page, select the name of the server to which you want to add a library.
   The File Systems Summary page is displayed.
2. Expand the Archive Media section and select Tape Libraries.
   The Tape Library Summary page is displayed.
3. Click Add to start the Add a Library wizard.
4. Follow the steps to add the device.
   When you have completed the wizard steps, the `samst.conf` file is automatically updated.

How To Configure Device Support for a Direct Attached Library

1. Copy the `/kernel/drv/samst.conf` file to a backup file. For example:

   ```
   # cp /kernel/drv/samst.conf /kernel/drv/samst.conf.orig
   ```

2. Edit the `/kernel/drv/samst.conf` file.
3. To include SCSI-attached magneto-optical drives or SCSI-attached libraries, do the following:
   a. Search for entries that have the following format to locate the list of SCSI targets and LUNs:

      ```
      name="samst" class="scsi" target=<target> lun=<lun>;
      ```

      target is the target ID for each SCSI drive found.
      lun is the corresponding LUN for each SCSI drive found.

   b. Within the list, find the entry that corresponds to each SCSI target ID and LUN on your inventory list.
      For example, the StorageTek 9730 automated library is attached to target 0 and LUN 0. The following line corresponds to that interface:

      ```
      name="samst" class="scsi" target=0 lun=0;
      ```

      Note that an entry might extend over two lines if it contains return characters.

      - If the entry starts with a pound character (#), delete the character. A pound (#) character marks a line as a comment.
      - If the entry is missing, create an entry for the SCSI target and LUN line you need, following the format shown in Step a and using the information in your hardware inventory list.
4. To include FC-attached magneto-optical drives or FC-attached automated libraries, create a line for each FC-attached one in your inventory list. Place the lines at the end of the SCSI target and LUN list, using the following format:

   ```
   name="samst" parent="fp" lun=<lun> fc-port-wwn="<world-wide-name>"
   ```

   For lun, specify the LUN for the drive.
   For world-wide-name, specify the WWN for the drive.
   The following example shows the line added to support the StorageTek L700 tape drive in the Example Inventory List.

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

5. Save your changes and exit the `samst.conf` file.

Verifying That All Devices Are Configured

1. Use the `cfgadm`(1M) command to list the devices included in the SAM environment. For example:

```
# cfgadm -al
Ap_Id                          Type         Receptacle   Occupant     Condition
c0                             scsi-bus     connected    configured   unknown
c0::dsk/c0t6d0                 CD-ROM       connected    configured   unknown
c1                             fc-private   connected    configured   unknown
c1::500000e0103c3a91           disk         connected    configured   unknown
c2                             scsi-bus     connected    unconfigured unknown
c3                             scsi-bus     connected    unconfigured unknown
c4                             scsi-bus     connected    configured   unknown
c4::dsk/c4t1d0                 disk         connected    configured   unknown
c4::dsk/c4t2d0                 disk         connected    configured   unknown
c5                             fc-fabric    connected    configured   unknown
c5::100000e00222ba0b           disk         connected    unconfigured unknown
c5::210000e08b0462e6           unknown      connected    unconfigured unknown
c5::210100e08b2466e6           unknown      connected    unconfigured unknown
c5::210100e08b27234f           unknown      connected    unconfigured unknown
c5::500104f00043abfc           tape         connected    configured   unknown
c5::500104f00043bc94           tape         connected    configured   unknown
c5::500104f00045eeaf           tape         connected    configured   unknown
c5::500104f000466943           tape         connected    configured   unknown
c5::500104f00046b3d4           tape         connected    configured   unknown
c5::500104f0004738eb           tape         connected    configured   unknown
c6                             fc           connected    unconfigured unknown
c7                             scsi-bus     connected    unconfigured unknown
c8                             scsi-bus     connected    unconfigured unknown
usb0/1                         usb-kbd      connected    configured   ok
usb0/2                         usb-mouse    connected    configured   ok
usb0/3                         unknown      empty        unconfigured ok
usb0/4                         unknown      empty        unconfigured ok
```

2. Examine the output to make sure that it shows all the devices you want configured in your SAM environment.
If a device is not configured, use the `cfgadm`(1M) command to configure it. For more information, see the `cfgadm`(1M) man page.
Because of a bug in the `cfgadm`(1) command, you might receive an error similar to the following:

```
# cfgadm -c configure -o force_update c4::500104f000489fe3
cfgadm: Library error: failed to create device node: 500104f00043abfc: Device busy
```

Despite the error, the `cfgadm`(1M) command processes the request.

Enable Storage Device Configuration

You must reboot the system to have the changes you have made to the `st.conf` and `samst.conf` files take effect.

```
# reboot -- -r
```

Handling Errors in the **`st.conf`** File

Errors can occur if the `st.conf` file is not configured properly during SAM software installation.

The following messages in the `sam-log` file indicate that the appropriate changes have not been made to `/kernel/drv/st.conf`. Follow the steps in How To Add Tape Devices to the `/kernel/drv/st.conf` File to fix the error.

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

The following device log messages correspond to the sam-log message:

```
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
```

Configuring the File System Environment

Each SAM-QFS software environment is unique. The system requirements and hardware differ from site to site. SAM-QFS environments support a wide variety of tape and optical devices, automated libraries, and disk drives. The system administrator at your site must set up the specific configuration for your environment.

The master configuration file, `/etc/opt/SUNWsamfs/mcf`, defines the equipment topology managed by the SAM-QFS software. This file specifies the devices, automated libraries, and file systems included in the environment. You assign each piece of equipment a unique Equipment Identifier in the `mcf` file.

You can edit the `mcf` file in either of two ways:

- By using the SAM-QFS Manager interface to configure archiving and file system devices. When you create a file system using SAM-QFS Manager, it creates an `mcf` file in `/etc/opt/SUNWsamfs/mcf` that contains a line for each device and family set of the file system.
- By directly editing the `mcf` file using a text editor.

The `mcf` file has two kinds of entries:

- File system device entries for disk devices. In the `mcf` file, you organize them into one or more file systems.
- Removable media device entries that you can organize into family sets. The `mcf` file contains information that enables you to identify the drives to be used and associate them with the automated libraries to which they are attached.

For detailed information about `mcf` file structures and contents, see About the Master Configuration File.

Example `mcf` files are installed in `/opt/SUNWsamfs/examples`. Several example `mcf` file configurations are also provided in Examples of `mcf` Files.

The following sections provide examples and describe activities related to creating and maintaining the `mcf` file.

How to Create the Master Configuration File (mcf) Manually

- Use vi(1) or another editor to create the `/etc/opt/SUNWsamfs/mcf` file.
  For detailed information about the contents of the `mcf` file, see About the Master Configuration File.

  > You can copy an example `mcf` file from `/opt/SUNWsamfs/examples` or from the examples in Examples of `mcf` Files.

  When you create the `mcf` file, follow these guidelines:

  - Delimit the fields in each line with spaces or tabs.
  - Begin each comment line entered into this file with a pound sign (#).
  - Use a dash (-) to indicate optional fields that are omitted.
    The following example shows the `mcf` file fields.

```
#
# Sun  Storage Archive Manager file system configuration
#
# Equipment       Equip  Equip Fam   Dev    Additional
# Identifier      Ord    Type  Set   State  Parameters
# ----------      -----  ----- ----  -----  ----------
```

The mcf file can contain both comment lines and lines that pertain to a device. The types of lines that can pertain to a device are as follows:

- Family set parent identifiers and family set devices
- Family set member devices
- Stand-alone devices

Identifying Peripherals Using the /var/adm/messages File

When your system boots, a series of messages is written to /var/adm/messages. These messages identify the Sun Solaris hardware path to each of the peripherals on your system. You can use this information to create the mcf file. To display information from the latest system reboot, search backward from the end of the file.

As the following example shows, each SCSI peripheral has three lines. The sixth field, samst2, indicates that these lines are associated with each other.

Example – SCSI Peripheral Lines in the **/var/adm/messages** File

```
# tail -200 /var/adm/messages | more
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@5,8800000/samst@2,0
```

- The first line displays the vendor and product information that the SCSI peripheral reported to the Sun Solaris kernel.
- The second line displays the SCSI bus, SCSI target ID, and LUN of the peripheral.
- The third line displays the peripheral's hardware path. This path is reflected in the /devices directory. Symbolic links (symlinks) to the /devices directory are set up in the /dev/st, /dev/samst, and /dev/rmt directories. Note that the third line wraps to the next line.

Matching the symbolic link to the correct peripheral is the key to configuring a Sun Storage Archive Manager environment. Use the ls(1) command with the -l option in both the /dev/st, /dev/samst and /dev/rmt directories to display the path name of the peripheral.

You might also want to set up the "device down" notification script at this point. The dev_down.sh(1M) man page contains information about setting up this script, which sends email to root when a device is marked down or off. For more information, see the dev_down.sh(1M) man page.

How to Verify the mcf File

- If you created your mcf file manually, use the sam-fsd(1M) command to verify the file.
  If you created your mcf file using SAM-QFS Manager, you do not need to verify its syntax.
  If the mcf file is free of syntax errors, the sam-fsd(1M) output includes information about the file systems, archiving, and other system information. If the mcf file contains syntax or other errors, however, the output is similar to the following example.

```
# sam-fsd
13: /dev/dsk/c1t1d0s0   10      md      samfs1  on      /dev/rdsk/c1t1d0s0
*** Error in line 13: Equipment name '/dev/dsk/c1t1d0s0' already in use by eq 10
72: /dev/rmt/3cbn       45      ug      l1000   on
*** Error in line 72: Equipment name '/dev/rmt/3cbn' already in use by eq 44
2 errors in '/etc/opt/SUNWsamfs/mcf'
sam-fsd: Read mcf /etc/opt/SUNWsamfs/mcf failed.
```

If the mcf file has errors, return to Setting Up the Environment Configuration and refer to the mcf(4) man page for information about creating this file. You can also refer to Examples of mcf Files.

How to Create an mcf File Using SAM-QFS Manager

Before You Begin

When you configure QFS file systems using the SAM-QFS Manager software, it creates or edits the appropriate configuration files, including the mcf file, on that server. You can use either SAM-QFS Manager or the CLI to further edit these files later.

> Note -
> If you want to use SAM-QFS Manager to configure your archiving environment and you want to include network attached libraries (excluding STK Libraries) in this configuration, you must create your parameters file before you create the mcf file. For information about creating a parameters file, see Creating Parameters Files for Network Attached Automated Libraries.
> You can add a StorageTek ACSLS network library in the SAM-QFS Manager without creating the parameters file. The application automatically generates the parameters file for you when you add the library in the Library Summary Page.

Steps

1. From your web browser, log in to the SAM-QFS Manager as an administrator user.
2. Expand the Getting Started section and choose First Time Configuration.
3. In section 2, click Create a File System.
   The New File System wizard is displayed.
4. Follow the steps for creating a new file system.
   When you have completed this process, the mcf file is created. For more information, see the SAM-QFS Manager online help.

Setting Up Mount Parameters

Use the procedures in this section to specify mount parameters for the file system.

You can specify mount parameters in the following ways:

- Using the mount(1M) command. Mount options specified here override those specified in the /etc/vfstab file and in the samfs.cmd file.
- In the /etc/vfstab file. Mount options specified here override those specified in the samfs.cmd file.
- In the samfs.cmd file.

For a list of available mount options, see the mount_samfs(1M) man page.

Updating the /etc/vfstab File and Creating the Mount Point

This section describes how to edit the /etc/vfstab file.

The following table shows the values you can provide in the fields in the /etc/vfstab file.

Table – **/etc/vfstab** File Fields

Field	Field Title and Content
1	Device to Mount. The name of the file system to be mounted. This value must be the same as the file system's Family Set name specified in the mcf file.
2	Device to fsck(1M). Must be a dash (-) character, which indicates that there are no options. This character prevents the Solaris system from performing an fsck(1M) process on the file system. For more information about this process, see the fsck(1M) or samfsck(1M) man page.
3	Mount Point, for example, /samfs1.
4	File System Type. Must be samfs.

5	`fsck`(1M) Pass. Must be a dash (`-`) character, which indicates that there are no options.
6	Mount at Boot. Either yes or no. • Specifying `yes` in this field indicates that the Sun Storage Archive Manager file system is to be mounted automatically at boot time. • Specifying no in this field indicates that you do not want to mount the file system automatically. For information about the format of these entries, see the `mount_samfs`(1M) man page.
7	Mount Parameters. A list of comma-separated parameters (with no spaces) that are used in mounting the file system. You can specify mount options on the `mount`(1M) command, in the `/etc/vfstab` file, or in a `samfs.cmd` file. Mount options specified on the `mount`(1M) command override those specified in the `/etc/vfstab` file and in the `samfs.cmd` file. Mount options specified in the `/etc/vfstab` file override those in the `samfs.cmd` file. For a list of available mount options, see the `mount_samfs`(1M) man page.

When you create a file system using SAM-QFS Manager, a default `/etc/vfstab` file is created. However, mount options specified in SAM-QFS Manager are written to the `samfs.cmd` file rather than to the `/etc/vfstab` file. For more information, see Creating and Editing the `samfs.cmd` File.

To edit the mount options in the `/etc/vfstab` file, use the following procedure, How to Update the /etc/vfstab File and Create the Mount Point Using a Text Editor.

How to Update the /etc/vfstab File and Create the Mount Point Using a Text Editor

The example in this task assumes that `/samfs1` is the mount point of the `samfs1` file system.

1. In the `/etc/vfstab` file, create an entry for each file system.
   The following example shows header fields and entries for a local file system.

```
#DEVICE    DEVICE   MOUNT   FS    FSCK  MOUNT    MOUNT
#TO MOUNT  TO FSCK  POINT   TYPE  PASS  AT BOOT  PARAMETERS
#
samfs1     -        /samfs1 samfs  -    yes      high=80,low=60
```

2. Use the `mkdir`(1M) command to create the mount point.
   For example:

```
# mkdir /samfs1
```

Creating and Editing the samfs.cmd File

You can create the `/etc/opt/SUNWsamfs/samfs.cmd` file as the place from which the system reads mount parameters. If you are configuring multiple file systems with multiple mount parameters, consider creating this file.

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

How to Create and Edit the samfs.cmd File Using SAM-QFS Manager

If you specify non-default mount options when creating a file system in SAM-QFS Manager, the `samfs.cmd` file is created or updated with those mount options.

Follow these steps to edit a file system's mount options:

1. On the Managed Hosts page, click the name of the server on which the file system is located.
   The File Systems Summary page is displayed.
2. Select the radio button next to the file system whose mount options you want to edit.
3. From the Operations menu, choose Edit Mount Options.
   The Edit Mount Options page is displayed.
4. Make your edits in the fields.
   For more information about the fields on the Edit Mount Options page, see the File System Manager online help.
5. Click Save.

The new mount options are written to the samfs.cmd file.

How to Create and Edit the samfs.cmd File Using a Text Editor

- Use vi(1) or another editor to create the samfs.cmd file.
  Create lines in the samfs.cmd file to control mounting, performance features, or other aspects of file system management. For more information about the samfs.cmd file, see The samfs.cmd File or the samfs.cmd(4) man page.

Initializing the Environment

This section tells you how to initialize the environment and the file system, and how to mount the file system.

How to Initialize the Environment

1. Use the samd(1M) config command to initialize the archiving and file system environment.
   For example:

```
# samd config
```

How to Initialize the File System

This procedure describes how to use the sammkfs(1M) command and the Family Set names that you have defined to initialize a file system.

> Note -
> The sammkfs(1M) command sets one tuning parameter, the disk allocation unit (DAU). You cannot reset this parameter without reinitializing the file system. For information about how the DAU affects tuning, see File Allocation Methods and the sammkfs(1M) man page.

1. Use the sammkfs(1M) command to initialize a file system for each Family Set name defined in the mcf file.

   > Caution -
   > Running the sammkfs(1M) command creates a new file system. It removes all references to the data currently contained in the partitions associated with the file system in the /etc/opt/SUNWsamfs/mcf file.

   The following example shows the command to initialize a file system with the Family Set name of samfs1.

```
# sammkfs samfs1
sammkfs: Configuring file system
Building "samfs1" will destroy the contents of devices:
    /dev/dsk/c2t0d0s3
    /dev/dsk/c2t0d0s7
Do you wish to continue? [y/N] *y*
total data kilobytes       = 16777728
total data kilobytes free  = 16777152
#
```

   The actual numbers returned vary from file system to file system.

Mounting the File System

The mount(1M) command mounts a file system and reads the /etc/vfstab and samfs.cmd configuration files. For information, see the mount_samfs(1M) man page.

How to Mount the File System Using SAM-QFS Manager

1. On the Managed Hosts page, click the name of the server on which the file system is located.
   The File Systems Summary page is displayed.
2. Select the radio button next to the file system that you want to mount.
3. From the Operations menu, choose Mount.

How to Mount the File System From the Command Line

1. Use the mount(1M) command to mount the file system.
   Specify the file system mount point as the argument. For example:

   ```
   # mount /samfs1
   ```

   Tip -
   If the file system has not been added to the /etc/vfstab file, use the following form of the mount command:

   ```
   #mount -F samfs <fs_name> </mount_point>
   ```

2. Use the mount(1M) command with no arguments to verify the mount.
   This step confirms that the file system is mounted and shows how to set permissions. The following example shows the output from a mount(1M) command issued to verify whether example file system samfs1 is mounted.

   ```
   # mount
   _<<< information deleted >>>_
   /samfs1 on samfs1 read/write/setuid/intr/largefiles/onerror=panic/dev=8001e3 on Thu Feb  5
   11:01:23 2004
   _<<< information deleted >>>_
   ```

3. (Optional) Issue the chmod(1) and chown(1) commands to change the permissions and ownership of the file system's root directory.
   Typically, this step is performed if this is the first time that the file system has been mounted. For example:

   ```
   # chmod 755 /samfs1
   # chown root:other /samfs1
   ```

The page Upgrading SAM and QFS does not exist.

Uninstalling the SAM-QFS Manager Software

For instructions on uninstalling the Sun QFS or SAM-QFS packages, see Removing the Existing Software.

How to Uninstall the SAM-QFS Manager Software

1. Log in to the server on which SAM-QFS Manager software is installed.
   This is the host on which you ran the fsmgr_setup script at installation time.
2. Become superuser.
3. Issue the following command to remove the SAM-QFS Manager software and all the applications that were installed with it:

```
# /var/sadm/samqfsui/fsmgr_uninstall
```

This script prompts you to confirm removal of the Tomcat Web Server, JRE packages, and information pertaining to administrator and user accounts.

End of Complete (Printable) SAM-QFS 5.1 Installation Guide

Complete (Printable) Sun QFS 5.1 Installation Guide

Sun QFS 5.1 Installation Guide

Contents

Preparing for Installation

This section explains the system requirements for the Sun QFS and SAM-QFS products and the tasks you must complete before you begin to install and configure your software.

Hardware and Software Requirements

You can install the software either on a Sun server based on UltraSPARC ® technology or on a server based on AMD Opteron x64 technology.

Additional requirements for the server that you want to use as the web server host for the SAM-QFS Manager browser interface tool are described in SAM-QFS Manager Requirements.

The software package runs on many Sun workstations and servers. Before installation, you should verify the compatibility of the hardware and the level of the Solaris Operating System (OS).

Operating System Requirements

Before installation, verify the applicability of the hardware and the level of the operating system. To install the software, you also must have root-level access to your system.

Sun Storage Archive Manager and Sun QFS 5.1 software require the following minimum operating system release:

- Solaris 10, Update 6

In addition, you can use any of the following operating systems as a client in a shared file system:

- Solaris 10 OS for x86 (32-bit)
- Red Hat Enterprise Linux 4.0 (UD-4) for x64 platforms
- Red Hat Enterprise Linux 4.5 for x64 platforms
- SuSE Linux Enterprise Server 9 (service pack 2) for x64 platforms
- SuSE Linux Enterprise Server 10 for x64 platforms
- SuSE Linux Enterprise Server 10 (service pack 2) for x64 platforms

How to Verify the Environment

Repeat these steps for each host on which you want to install the software.

1. Verify that your system has a CD-ROM drive or that it can access the release package at the Sun Download Center
2. Log in to your system as `root`.
   You must have superuser access to install the software.
3. Verify your system's Solaris OS level.
   For example, the output from the following command will show the major and minor OS release information as well as the architecture:

   ```
   % cat /etc/release
   ```

   The software relies on properly configured Solaris software at the following minimum release level:

   - Solaris 10, Update 6

Installing Solaris OS Patches

Sun Microsystems provides Solaris OS patches to customers with a maintenance contract by means of CD-ROM, anonymous FTP, and the Sun Microsystems SunSolve™ web site .

To install a patch after you install the Sun QFS or SAM-QFS release packages, load 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 included in the patch or jumbo patch cluster.

Software Host Requirements

If you plan to install the software in a multihost environment, such as a SAM-Remote configuration, all host systems must have the same software release level installed and operational.

Verifying Shared File System Requirements

This section describes the system requirements for a Sun QFS shared file system.

Metadata Server Requirement

You must have at least one Solaris metadata server. If you want to be able to change the metadata server, you must have at least one other Solaris host that can become the metadata server. These additional host systems are known as potential metadata servers. These servers must all be running on the same hardware platform, either SPARC or x64. You cannot mix server hardware platforms. In a Sun Cluster environment, all nodes included in a shared file system are potential metadata servers.

The following are configuration recommendations with regard to metadata storage:

- A shared file system should have multiple metadata (`mm`) partitions. This spreads out metadata I/O and improves file system throughput.
- A shared file system should use a separate, private metadata network so that typical user traffic does not interfere with metadata traffic. A switch-based (not hub-based) network is recommended.

Operating System and Hardware Requirements

Ensure that your configuration meets the following operating system and hardware requirements:

- The host systems to be configured in the Sun QFS shared file system must be connected by a network.
- All metadata servers and potential metadata servers must have the same processor type.
- The client systems can be installed on the Solaris OS or on one of the following operating systems:
    - Red Hat Enterprise Linux 4.0 (UD-4) for x64 platforms (Sun QFS shared client only)
    - Red Hat Enterprise Linux 4.5 for x64 platforms (Sun QFS shared client only)
    - SuSE Linux Enterprise Server 9 (service pack 2) for x64 platforms (Sun QFS shared client only)
    - SuSE Linux Enterprise Server 10 for x64 platforms (Sun QFS shared client only)
    - SuSE Linux Enterprise Server 10 (service pack 2) for x64 platforms (Sun QFS shared client only)
- Online data storage devices must be directly accessible to all hosts. All online metadata storage devices must be directly accessible to all potential metadata server hosts.

Sun Storage Archive Manager and Sun QFS Release Levels

Ensure that your configuration meets the following requirements:

- Each host to be configured in the shared file system must have the same software package installed.
- All software installed on the systems in the shared file system must be at the same release level. For example, if one host has the SAM-QFS 5.1 packages, all hosts that are part of the shared file system must have the SAM-QFS 5.1 packages.
  This requirement ensures that all systems in a shared file system have identical over-the-wire protocol versions. If these levels do not match, the system writes the following message to the metadata server's `/var/adm/messages` file when mounting is attempted:

  ```
  SAM-FS: <client> client package version <x> mismatch, should be <y>.
  ```

- When applying patches or upgrading the software for a shared file system, make sure to apply the same patch to all hosts that have access to the shared file system. Unexpected results might occur if not all host systems are running the same patch level.

Verifying Third-Party Compatibilities

The SAM-QFS software interoperates with many different hardware and software products from third-party vendors. Depending on your environment, you might need to upgrade other software or firmware before installing the SAM-QFS package. Consult the Release Notes for information pertaining to library model numbers, firmware levels, and other compatibility information.

SAM-QFS Manager Requirements

The SAM-QFS Manager browser interface is used to configure, control, monitor, or reconfigure a SAM-QFS environment using a graphical web browser interface.

You can install the SAM-QFS Manager software in one of the following configurations:

- As a stand-alone management station to manage one or more Sun QFS hosts
- As additional software on the Sun QFS host

After the SAM-QFS Manager software is installed, you can invoke the SAM-QFS Manager from any machine on the network that is allowed access to its web server.

For information about the requirements for the host upon which you are configuring the SAM-QFS Manager software, see Verifying Requirements for SAM-QFS Manager.

Determining Disk Space Requirements

The SAM-QFS software packages require a certain amount of disk cache (file system devices) to create and manage data files and directories.

Planning Your File System and Verifying Disk Cache

A local file system requires only a single partition. If you install SAM-QFS to enable archiving support, the file system requires either one or two partitions:

- To store file data separately from file system metadata (`ma` file system), you need to have at least two disk devices or partitions.
- To store data and metadata on the same device (`ms` file system), you need to have one disk device or partition.

The disk devices or partitions do not require any special formatting. You might see better performance if you configure multiple devices across multiple interfaces (HBAs) and disk controllers.

> Caution -
> Make sure that the disks and partitions that you plan to use are not currently in use and do not contain any existing data. Any existing data will be lost when you create the Sun QFS file system.

The disks must be connected to the server through a Fibre Channel (FC) or SCSI controller. You can specify individual disk partitions for a disk, or you can use the entire disk as a disk cache. The software supports disk arrays, including those under the control of volume management software, such as Solaris Volume Manager.

Before creating your first file system, you should familiarize yourself with file system layout possibilities. For information about volume management, file system layout, and other aspects of file system design, see the Sun QFS File System Configuration and Administration Guide.

> Note -
> If you are using a shared file system configuration that contains the Solaris 10 OS on both x64 platforms and SPARC platforms, Extensible Firmware Interface (EFI) labels are required on all shared disks. See Configuring EFI Labels for Shared x64 and SPARC Volumes for information about relabeling disks.

How to Estimate Disk Cache Requirements

Use the following guidelines to estimate the disk cache needed for SAM-QFS software (file systems plus the storage and archive manager):

- Disk cache = largest file (in bytes) + amount of space needed for working files
- Metadata cache
  Use the following information to estimate the metadata cache requirements. The metadata cache must have enough space to contain the following data:
  - Two copies of the superblock (16 Kbytes each)
  - Reservation maps for metadata space plus data space((metadata + file data)/disk allocation unit (DAU)/32,000) * 4 Kbytes
  - Inode space(number of files + number of directories) * 512 bytes
  - Indirect blocks – a minimum of 16 Kbytes each
  - Directory data space(number of directories * 16 Kbytes)

Run the `format`(1M) command to verify that you have sufficient disk cache space.

The `format`(1M) command shows how the disks are partitioned and the size of each partition.

Example 1 - Using the format(1M) Command on Fibre-Channel-Attached Disks

This example shows six disks attached to a server. Two internal disks are connected by means of controller 0 on targets 10 and 11 (`c0t10d0` and `c0t11d0`). The other disks are external.

For the sake of clarity, the `format`(1M) command output in this example has been edited.

Example – **`format`**(1M) Command for Fibre-Channel-Attached Disks

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
       0. c0t10d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /sbus@3,0/SUNW,fas@3,8800000/sd@a,0
       1. c0t11d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /sbus@3,0/SUNW,fas@3,8800000/sd@b,0
       2. c9t60020F2000003A4C3ED20F150000DB7Ad0 <SUN-T300-0118 cyl 34530 alt 2 hd 48 sec 128>
          /scsi_vhci/ssd@g60020f2000003a4c3ed20f150000db7a
       3. c9t60020F2000003A4C3ED215D60001CF52d0 <SUN-T300-0118 cyl 34530 alt 2 hd 48 sec 128>
          /scsi_vhci/ssd@g60020f2000003a4c3ed215d60001cf52
       4. c9t60020F2000003A4C3ED21628000EE5A6d0 <SUN-T300-0118 cyl 34530 alt 2 hd 48 sec 128>
          /scsi_vhci/ssd@g60020f2000003a4c3ed21628000ee5a6
       5. c9t60020F2000003A4C3ED216500009D48Ad0 <SUN-T300-0118 cyl 34530 alt 2 hd 48 sec 128>
          /scsi_vhci/ssd@g60020f2000003a4c3ed216500009d48a
Specify disk (enter its number):^d
#
# format /dev/rdsk/c9t60020F2000003A4C3ED216500009D48Ad0s2
# *format f*
partition> p


Part      Tag    Flag     Cylinders         Size            Blocks
  0 unassigned    wm       0 -  4778       14.00GB    (4779/0/0)   29362176
  1 unassigned    wm    4779 -  9557       14.00GB    (4779/0/0)   29362176
  2     backup    wu       0 - 34529      101.16GB    (34530/0/0) 212152320
  3 unassigned    wm    9558 - 14336       14.00GB    (4779/0/0)   29362176
  4 unassigned    wm   14337 - 19115       14.00GB    (4779/0/0)   29362176
  5 unassigned    wm   19116 - 23894       14.00GB    (4779/0/0)   29362176
  6 unassigned    wm   23895 - 28673       14.00GB    (4779/0/0)   29362176
  7 unassigned    wm   28674 - 33452       14.00GB    (4779/0/0)   29362176

partition> ^D
#
```

Example 2 - Using the format(1M) Command on SCSI-Attached Disks

The following example shows four disks attached to a server. Two internal disks are connected by means of controller `0` on targets `0` (`c0t0d0`) and `1` (`c0t1d0`). Two external disks are connected by means of controller `3` on targets `0` (`c3t0d0`) and `2` (`c3t2d0`).

Example – **`format`**(1M) Command for SCSI-Attached Disks

```
# format
Searching for disks...done

AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
          /sbus@1f,0/SUNW,fas@e,8800000/sd@0,0
       1. c0t1d0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
          /sbus@1f,0/SUNW,fas@e,8800000/sd@1,0
       2. c3t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /sbus@1f,0/QLGC,isp@0,10000/sd@0,0
       3. c3t2d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /sbus@1f,0/QLGC,isp@0,10000/sd@2,0
Specify disk (enter its number): *1*
selecting c0t1d0
[disk formatted]
Warning: Current Disk has mounted partitions.

FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        repair     - repair a defective sector
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        inquiry    - show vendor, product and revision
        volname    - set 8-character volume name
        <cmd>      - execute <cmd>, then return
        quit
format> par

PARTITION MENU:
        0      - change "0" partition
        1      - change "1" partition
        2      - change "2" partition
        3      - change "3" partition
        4      - change "4" partition
        5      - change "5" partition
        6      - change "6" partition
        7      - change "7" partition
        select - select a predefined table
        modify - modify a predefined partition table
        name   - name the current table
        print  - display the current table
        label  - write partition map and label to the disk
        <cmd>  - execute <cmd>, then return
        quit
partition> pri
Current partition table (original):
Total disk cylinders available: 2733 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0        var    wm       0 - 2732        1.98GB    (2733/0/0) 4154160
  1 unassigned    wm       0               0         (0/0/0)          0
  2     backup    wm       0 - 2732        1.98GB    (2733/0/0) 4154160
  3 unassigned    wm       0               0         (0/0/0)          0
  4 unassigned    wm       0               0         (0/0/0)          0
  5 unassigned    wm       0               0         (0/0/0)          0
  6 unassigned    wm       0               0         (0/0/0)          0
  7 unassigned    wm       0               0         (0/0/0)          0

partition> q
```

Verifying Disk Space

The software requires a disk cache consisting of redundant arrays of inexpensive disks (RAID) devices, JBOD ("just a bunch of disks") devices, or both. It also requires a certain amount of disk space in the `/` (root), `/opt`, and `/var` directories. The actual amount needed varies depending on the packages you install.

The following table shows the minimum amount of disk space required in these various directories.

Table – Minimum Disk Space Requirements

Directory	SAM (Archiving and File System)	QFS (File System Only)	SAM-QFS Manager
`/` (root) directory	2 Mbytes	2 Mbytes	25 Mbytes
`/opt` directory	21 Mbytes	8 Mbytes	5 Mbytes
`/var` directory	4 Gbytes	1 Mbyte	2 Mbytes
`/usr` directory	2 Mbytes	2 Mbytes	7 Mbytes
`/tmp` directory	0 Mbytes	0 Mbytes	200 Mbytes

Note -
The space requirements for the `/var` directory take into account the fact that the archiver data directory, the archiver queue files, and the log files are written to the `/var` directory.

How to Verify Disk Space

The following procedure shows how to verify whether there is enough disk space on your system to accommodate the `SUNWsamfsu` and `SUNWsamfsr` software packages.

1. Run the following command to verify that at least 2 Mbytes are in the `avail` column for the `/` directory.

```
# df -k /
Filesystem        kbytes  used    avail capacity   Mounted on
/dev/dsk/c0t1d0s0 76767   19826   49271    29%     /
```

2. Run the following command to verify that at least 21 Mbytes are in the `avail` column for the `/opt` directory.

```
# df -k /opt
Filesystem        kbytes  used    avail capacity   Mounted on
/dev/dsk/c0t1d0s4 192423  59006  114177    35%     /opt
```

3. Verify that at least 6 Mbytes are available in the `/var` directory.
   A quantity of 30 Mbytes or more is recommended to allow for the growth of log files and other system files.
4. If each directory does not have enough room for the software, repartition the disk to make more space available to each file system.
   For information about how to repartition a disk, see your Sun Solaris system administration documentation.

Preparing Hardware for Archiving

Verifying Archive Media

If you plan to perform disk archiving (to archive to disk space in another file system), verify the following:

- The host system to which the disks are attached has at least one file system created that is compatible with the SAM-QFS software.
- The disk has enough space available to accommodate the archive copies.

If you plan to archive to removable media devices, your environment must include the following:

- At least one removable media device for archiving files. This device can be a single tape or optical drive, or it can be multiple devices, such as the drives within an automated library.
- Tape or magneto-optical cartridges to which archive files can be written. For most SCSI-attached and FC-attached libraries, the

SAM-QFS software supports only one media type. If you have a tape library that can be partitioned logically into two or more libraries, you can have one media type in one logical library and a different media type in another. The SAM-QFS software records the cartridges used for each library in a library catalog. You cannot mix the tape media types in a library catalog, so plan to use only one media type per library or logical library.

The SAM-QFS environment supports a wide variety of removable media devices. You can obtain a list of currently supported drives and libraries from your Sun Microsystems sales or support staff.

To make sure that your devices are attached and enumerated in an easily retrieved list, perform one or both of the following procedures:

- If your removable media devices are not attached to your server, perform the procedure in Verifying Archive Media.
- Enumerate your devices using the instructions in Creating a List of Devices. You will use this list again in Installing the Software Packages.

How to Attach Removable Media Devices

The following steps are general guidelines for attaching removable media hardware to a server. For explicit instructions on how to connect these peripherals to a server, refer to the hardware installation guide supplied by the vendor with the automated library and drives.

1. Ensure that you are on a console connection to the server.
2. Power off the server.
3. Ensure that the removable media devices and the disks to be used for the Sun QFS file system are connected and properly addressed.
4. If you have libraries attached to the host system through a SCSI interface, ensure that the SCSI target IDs are unique for each SCSI initiator (host adapter).
   Avoid setting SCSI target IDs for peripherals to IDs that are already in use. In addition, if you are using a SCSI host adapter with a previously attached disk drive, any additional peripheral connected to this bus must have a different ID. Typically, the initiator uses ID 7, and the internal disk drive uses ID 3 for SPARC systems and ID 0 for UltraSPARC systems.
5. Power on the peripherals according to the manufacturer's recommended sequence.
   Typically, you power on the outermost peripherals first, working toward more central components in sequence.
6. Disable autobooting.
   At the `>ok` prompt, type the following command to disable autobooting:

   ```
   >ok setenv auto-boot? false
   ```

7. Type `reset` at the next prompt:

   ```
   >ok reset
   ```

8. Do one of the following:
   - If you have libraries attached to the host system through a SCSI interface, use the `probe-scsi-all` command to conduct an inventory of target IDs and logical unit numbers (LUNs) for each device connected to the host system. Save the output. You will use the information in this output for the next procedure, Creating a List of Devices.
     For example:

     ```
     {0} ok probe-scsi-all
     /pci@6,400/scsi@2,1
     Target 0
       Unit 0   Removable Device type 8     STK 9730        1700
     Target 1
       Unit 0   Removable Tape   type 7     QUANTUM DLT7000 2565
     Target 2
       Unit 0   Removable Tape   type 7     QUANTUM DLT7000 2565
     /pci@1f,4000/scsi@3
     Target 0
       Unit 0   Disk     SEAGATE ST318404LSUN18G 4207
     Target 6
       Unit 0   Removable Read Only device   TOSHIBA XM6201TASUN32XCD1103
     ```

   - If libraries or tape drives are attached to the host system through an FC interface, conduct an inventory of target IDs and LUNs for each device connected to the host system. Save the output. You will use the information in this output for the next procedure, Creating a List of Devices.
     For example:

```
{0} ok show-devs
/SUNW,ffb@1e,0
/SUNW,UltraSPARC-II@2,0
/SUNW,UltraSPARC-II@0,0
/counter-timer@1f,1c00
/pci@1f,2000
/pci@1f,4000
/virtual-memory
/memory@0,a0000000
/aliases
/options
/openprom
/chosen
/packages
/pci@1f,2000/SUNW,qlc@1
/pci@1f,2000/SUNW,qlc@1/fp@0,0
/pci@1f,2000/SUNW,qlc@1/fp@0,0/disk
/pci@1f,4000/SUNW,ifp@2
/pci@1f,4000/scsi@3,1
/pci@1f,4000/scsi@3
/pci@1f,4000/network@1,1
/pci@1f,4000/ebus@1
/pci@1f,4000/SUNW,ifp@2/ses
{0} ok select /pci@1f,2000/SUNW,qlc@1
{0} ok show-children
LiD HA LUN  --- Port WWN ---   ----- Disk description -----
 2  7e   0  500104f00041182b  STK     L700            0236
7c  7e   0  500104f00043abfc  STK     9840            1.28
7d  7e   0  500104f00045eeaf  STK     9840            1.28
6f  7e   0  500104f000416304  IBM     ULT3580-TD1     16E0
6e  7e   0  500104f000416303  IBM     ULT3580-TD1     16E0
```

If the server does not acknowledge all the known devices (disk drives, tape or optical drives, the automated library, and so on), check the cabling. Do not proceed until all devices appear when probed.

9. Reenable autobooting, and then boot the system:

```
>ok setenv auto-boot? true
>ok *boot*
```

10. Review system files.
    Review the following files:
    - `/var/adm/messages` to ensure that all devices were recognized
    - `/dev/rmt` for expected tape devices
    - `/dev/dsk` and `/dev/rdsk` for expected disks
      Due to special driver requirements, no device information appears in `/var/adm/messages` for magneto-optical devices or libraries until after you install the SAM-QFS software packages.
11. Disable autocleaning and autoloading.
    If your automated library supports autocleaning or autoloading, disable those features when using that library with the Sun Storage Archive Manager software. Consult the documentation from your library's manufacturer for information about disabling autocleaning and autoloading.

> Note -
> The only times you can use autoloading are during the initial loading of cartridges and when the SAM-QFS software is not running. Remember to disable autoloading when the SAM-QFS system is running.

Creating a List of Devices

The devices that you intend to use must be attached and recognized by the server upon which you intend to install the SAM-QFS software. To configure the SAM-QFS software, you need to know the following about your devices:

- The device type, manufacturer, and model number.
- The mechanism by which the device is attached to the server. You can attach devices in one of the following ways:

- Drives can use either a SCSI attachment or an FC attachment. Each drive accepts either tape cartridges or magneto-optical cartridges.
  For SCSI-attached drives, you need to know each drive's SCSI target ID and logical unit number (LUN).
  For FC-attached drives, you need to know each drive's LUN and node World Wide Name (WWN).
- Automated libraries can use a SCSI attachment, an FC attachment, or a network attachment.
  Libraries that use SCSI or FC attachments are called direct attached libraries. For SCSI-attached libraries, you need to know each library's SCSI target ID and LUN. For FC-attached libraries, you need to know each library's LUN and node WWN.
  Libraries that use a network attachment are called network attached libraries. You cannot configure network attached libraries in the existing system configuration files; instead, you must create a parameters file for each network attached library. This is explained later in the installation process.

How to Create a List of Devices

Fill in the following table to include the name, manufacturer, model, and connection types for each device that you want to include in your SAM-QFS environment. Retain this list for use again later in the configuration procedure.

Device Name, Manufacturer, and Model	Target ID	LUN	Node WWN
SCSI-attached tape drives			
			Not applicable
			Not applicable
			Not applicable
FC-attached tape drives			
	Not applicable		
	Not applicable		
	Not applicable		
SCSI-attached magneto-optical drives			
			Not applicable
			Not applicable
			Not applicable
FC-attached magneto-optical drives			
	Not applicable		
	Not applicable		
	Not applicable		
SCSI-attached automated libraries			
			Not applicable
			Not applicable
			Not applicable
FC-attached automated libraries			
	Not applicable		
	Not applicable		
	Not applicable		

Obtaining the Release Files

Make sure that you have a copy of the release software. You can obtain the Sun SAM-QFS software from the Sun Download Center or on a

CD-ROM. Contact your authorized service provider (ASP) or your Sun sales representative if you have questions on obtaining the software.

After the release, upgrade patches are available from Sun Solve.

> Caution -
> If you have not read the Release Notes, do so before you continue.

How to Obtain the Software From the Sun Download Center

1. Go to the downloads page.
2. Select the SAM-QFS or Sun QFS software package you want to receive.
3. Follow the instructions on the web site for downloading the software.

Software Licensing

You must agree to all binary and right-to-use (RTU) software license agreements before you install either the Sun QFS or the Sun SAM software.

Setting Up the Network Management Station

Perform this procedure if you want to monitor your configuration through Simple Network Management Protocol (SNMP) software.

You can configure the Sun SAM-QFS software to notify you when potential problems occur in its environment. The SNMP software manages information exchange between network devices such as servers, automated libraries, and drives. When the Sun SAM-QFS software detects potential problems in its environment, it sends information to a management station, which enables you to monitor the system remotely.

The management stations you can use include the following:

- Sun Storage Automated Diagnostic Environment (StorADE)
- Sun Management Center (Sun MC)
- Sun Remote Server (SRS)
- Sun Remote Services Net Connect

If you want to enable SNMP traps, make sure that the management station software is installed and operating correctly before installing the Sun SAM-QFS software. Refer to the documentation that came with your management station software.

The types of problems, or events, that the SAM-QFS software can detect are defined in the SAM-QFS Management Information Base (MIB). The events include errors in configuration, `tapealert`(1M) events, and other atypical system activity. For complete information about the MIB, see `/var/snmp/mib/SUN-SAM-MIB.mib` after the packages are installed.

The SAM-QFS software supports the TRAP SNMP (V2c) protocol. The software does not support `GET-REQUEST`, `GETNEXT-REQUEST`, and `SET_REQUEST`.

Release Package Contents, Directories, and Files

Release Package Contents

The Sun Storage Archive Manager (SAM-QFS) and Sun QFS software packages are in Sun Solaris `pkgadd`(1M) format. These packages reflect the Sun Solaris version for the platform on which you will be installing the software.

The following table shows the release package names.

Table – Release Package Names

Installed Package	Description
`SUNWqfsr` `SUNWqfsu`	Sun QFS (file system only) software packages

SUNWsamfsr SUNWsamfsu	SAM-QFS (archiving and file system) software packages
SUNWfsmgrr SUNWfsmgru	SAM-QFS Manager software packages; Run the `fsmgr_setup` script to install
SUNWsamfswm	WORM-FS support packages

The releases are identified using characters arranged in the following format:
major U update `.` patch

The U stands for "update" in this format.

In the patch number field, a number between 1 and 99 indicates a patch release, and a letter from A through Z indicates pre-release software. The base release of a first feature release of a major release might not contain a patch level.

For example:

- 4U0 is release 4, update 0, a major release with no minor release revisions and no bug fixes.
- 4U2 is release 4, update 2, a minor release.
- 4U2.1 is a patch release that contains software fixes for a major or minor release. This number appears in the patch's README file.

Directories and Files Created

This section describes the directories and files associated with the Sun QFS and SAM-QFS products. You can obtain additional information about the files in this section from the man pages after the software is installed.

Directories Created at Installation

The following table lists the directories created when the software packages are installed.

Table – Directories Created

Directory	Content
`/dev/samst`	Device driver special files (only when SAM-QFS packages are installed).
`/etc/fs/samfs`	Commands specific to the software.
`/etc/opt/SUNWsamfs`	Configuration files.
`/etc/opt/SUNWsamfs/scripts`	Site-customizable scripts.
`/opt/SUNWsamfs/bin`	User command binaries.
`/opt/SUNWsamfs/client`	Files for remote procedure call API client.
`/opt/SUNWsamfs/doc`	Documentation repository for any informational files included in the release. The README file, which summarizes the installed release's features, is included in this directory.
`/opt/SUNWsamfs/examples`	Various example configuration files.
`/opt/SUNWsamfs/include`	API include files.
`/opt/SUNWsamfs/lib`	Relocatable libraries.
`/opt/SUNWsamfs/man`	Man pages.
`/var/snmp/mib`	Standard MIB files and product MIB (`SUN-SAM-MIB.mib`).
`/opt/SUNWsamfs/sbin`	System administrator commands and daemon binaries.
`/opt/SUNWsamfs/sc`	Sun Cluster binaries and configuration files.
`/opt/SUNWfsmgr/bin`	SAM-QFS Manager administrator commands.
`/opt/SUNWfsmgr/doc`	SAM-QFS Manager online documentation repository.
`/var/opt/SUNWsamfs`	Device catalogs, catalog trace file, log files, and archiver data directory and queue files.

Files Created at Installation

The following table lists miscellaneous files created when the software is installed.

Table – Files Created - Miscellaneous

File	Description
`/etc/opt/SUNWsamfs/inquiry.conf`	Vendor and product identification strings for recognized SCSI devices (only when SAM-QFS packages are installed).
`/etc/sysevent/config/SUNW,SUNWsamfs,sysevent.conf`	Solaris system event handler configuration file.
`/kernel/drv/amd64/samaio`	File system asynchronous I/O pseudo-driver (64-bit version for x64 platforms).
`/kernel/drv/amd64/samioc`	Sun Solaris 64-bit file system interface module (for x64 platforms).
`/kernel/drv/amd64/samst`	SAM-QFS driver for SCSI media changers and optical drives for tape drives (64-bit version for x64 platforms).
`/kernel/drv/samaio.conf`	Configuration file for `samaio`.
`/kernel/drv/samioc.conf`	Configuration file for the `samioc` module.
`/kernel/drv/samst.conf`	Configuration file for the `samst` driver.
`/kernel/drv/sparcv9/samaio`	File system asynchronous I/O pseudo-driver (64-bit version for SPARC platforms).
`/kernel/drv/sparcv9/samioc`	Sun Solaris 64-bit file system interface module (for SPARC platforms).
`/kernel/drv/sparcv9/samst`	SAM-QFS driver for SCSI media changers and optical drives for tape drives (64-bit version for SPARC platforms).
`/kernel/fs/amd64/samfs`	Sun Solaris 64-bit file system module for the x64 platform.
`/kernel/fs/sparcv9/samfs`	Sun Solaris 64-bit file system module for SPARC platforms.
`/var/log/webconsole/host.conf`	SAM-QFS Manager configuration file.
`/var/opt/SUNWsamfs/faults`	Faults history file.
`/var/sadm/samqfsui/fsmgr_uninstall`	Software for removing SAM-QFS Manager and its supporting applications.
`/opt/SUNWsamfs/sc/etc/SUNW.qfs`	Sun Cluster configuration file created only in the presence of Sun Cluster software.
`/usr/cluster/lib/rgm/rtreg/SUNW.qfs`	Sun Cluster configuration file created only in the presence of Sun Cluster software.

The file system has dynamically loadable components that are stored in the Sun Solaris `/kernel` directory. You can use `modinfo`(1M) command to determine the modules that are loaded. Typically, the kernel loads the file system module at boot time. Alternatively, you can load the file system module when the file system is first mounted after the Sun software is installed.

Fault Notification Files

After the software is installed, it creates files that it uses for fault notification. The following table lists these files. When the software detects faults serious enough to merit user attention, the software uses these trap and log files to convey fault information through the SAM-QFS Manager software.

Table – Files Created - Fault Notification

File	Description
`/etc/opt/SUNWsamfs/scripts/sendtrap`	Sends trap information.

/opt/SUNWsamfs/sbin/fault_log	Records faults.
/opt/SUNWsamfs/sbin/tapealert_log	Records tapealert(1M) faults (only when SAM-QFS packages are installed).
/opt/SUNWsamfs/sbin/tapealert_trap	Sends tapealert(1M) traps (only when SAM-QFS packages are installed).

The software creates these files with the following permissions:

```
-rwxr-x---
```

Caution -
Do not change these file permissions.
If execute permissions are lost, for example, the system writes messages such as the following to /var/adm/messages:

```
SUNW,SUNWsamfs,sysevent.conf, line1: no execute access to
/opt/SUNWsamfs/sbin/tapealert_trap - No such file or directory.
```

Site Files

The configuration procedures elsewhere in this information direct you to create several site-specific files. The software uses these site files.

Note -
Your site's configuration files must contain ASCII characters only.

You must create the master configuration mcf file at your site in order to use the Sun SAM-QFS software. For more information about the /etc/opt/SUNWsamfs/mcf file, see About the Master Configuration File and the mcf(4) man page.

If you are using the archiver and file system features, you might also create all the files shown in the following table. If you are using only file system features, you might only create the first two files.

Table – Optional Site Files

File	Description
/etc/opt/SUNWsamfs/samfs.cmd	File system mount parameter command file. For more information, see the samfs.cmd(4) man page.
/etc/opt/SUNWsamfs/defaults.conf	Miscellaneous default values. For more information, see the defaults.conf(4) man page.
/etc/opt/SUNWsamfs/archiver.cmd	Archiver command file. For more information, see the archiver.cmd(4) man page, or Configuring the Archiver.
/etc/opt/SUNWsamfs/preview.cmd	Previewer command file. For more information, see Configuring the Stager and the preview.cmd(4) man page.
/etc/opt/SUNWsamfs/recycler.cmd	Recycler command file. For more information, see Configuring the Recycler and the recycler.cmd(4) man page.
/etc/opt/SUNWsamfs/releaser.cmd	Releaser command file. For more information, see About Releasing and the releaser.cmd(4) man page.

Modified System Files

During installation, the software adds information to certain Solaris system files. These system files are ASCII text files. The Solaris OS uses these files to identify loadable kernel modules by number rather than by name.

The software adds information to the following files:

- `/etc/name_to_major`
  The SAM-QFS software uses this file to map drivers to major numbers. The `samst` and `samrd` major numbers can vary, depending on the major numbers in use by the Solaris OS. The system adds the following lines to this file:

```
samst 63
samrd 64
samioc 236
samaio 237
```

- `/etc/security/auth_attr`
  This file is the authorization description database. The system adds the following lines to this file:

```
# File System Manager Authorizations
com.sun.netstorage.fsmgr.config:::File System Manager All Access::
com.sun.netstorage.fsmgr.operator.media:::File System Manager Media
Related Operation Access::
com.sun.netstorage.fsmgr.operator.sam.control:::File System Manager
Start/Stop/Idle Archiving Access::
com.sun.netstorage.fsmgr.operator.file:::File System Manager File
LevelOperation Access::
com.sun.netstorage.fsmgr.operator.filesystem:::File System Manager
FileSystem Level Operation Access::
```

- `/etc/user_attr`
  This file is the extended user attributes database used by SAM-QFS Manager.

```
root::::profiles=Web Console Management,All;auths=
Solaris.*,solaris.grant,*com.sun.netstorage.fsmgr.**;
lock_after_retries=no
```

- `/etc/inittab`
  The system adds the following line to this file:

```
sfad:3:respawn:/opt/SUNWsamfs/sbin/fsmgmtd
```

Installing Sun QFS

Complete the tasks below if this is the initial installation of the Sun QFS (file system only) software packages at your site.

- To upgrade Sun QFS software on an existing server, see Upgrading QFS and SAM-QFS.
- To install the software in a Sun Cluster environment, you must also follow the additional instructions in Using SAM-QFS With Solaris Cluster.

> Note
> You must be logged in as superuser to complete the installation tasks.
> Installing QFS involves many of the same steps as installing SAM-QFS, except that you do not configure the archiving storage devices.

Before You Begin

- If you are not already familiar with the QFS file system, read the File System Overview.
- Before you follow the detailed installation steps below, check the hardware and software requirements as explained in Preparing for Installation.

Installation Overview Task Map

Depending on the the features that you need to support, you must complete several of the following procedures.

Step	Task	Description
1	Add the software packages.	Install the appropriate packages for your needs. Also see Release Package Contents.
2	Configure `path` and `manpath` variables.	Configure the environment variables for access to commands and man pages.
3	(Optional) Install and configure SAM-QFS Manager.	This task is needed only if you want to use a browser to configure file systems.
4	Configure the file system environment.	Define the master configuration file `mcf`.
5	Configure file system mount parameters.	Define the `/etc/vfstab` and `samfs.cmd` files.
6	Initialize the environment.	Initialize SAM-QFS and mount the file systems.
7	(Optional) Configure shared file systems.	If applicable to your environment, complete the configuration tasks specific to a shared Sun QFS environment (see Configuring a Shared File System).
8	(Optional} Configure high-availability for your file systems.	If applicable to your environment, complete the configuration tasks specific to a Sun Cluster environment (see Using SAM-QFS With Solaris Cluster).

Installing the Software Packages

The Sun QFS and Sun Storage Archive Manager (SAM-QFS) software uses the Sun Solaris packaging utilities for adding and deleting software. The `pkgadd`(1M) utility prompts you to confirm various actions necessary to install the packages.

Table – Release Package Names

Installed Package	Description
`SUNWqfsr` `SUNWqfsu`	Sun QFS (file system only) software packages
`SUNWsamfsr` `SUNWsamfsu`	SAM-QFS(archiving and file system) software packages

How to Add the Packages

1. Become superuser.
2. Use the `cd` command to go to the directory where the software package release files reside.
   You obtained the release files as described in Obtaining the Release Files. Changing to the appropriate directory differs, depending on your release media, as follows:
   - If you downloaded the release files, go to the directory to which you downloaded the files.
   - If you obtained the release files from a CD-ROM, go to the directory on the CD-ROM that corresponds to your operating system version.
3. Use the `pkgadd`(1M) command to add the appropriate packages, based on the features that you need to support:
   - For archiving to a local or shared file system, install the `SUNWsamfsr` and `SUNWsamfsu` packages.
   - To support just a local or shared file system (no archiving), install the `SUNWqfsr` and `SUNWqfsu` packages.
     For example:

```
# pkgadd -d . SUNWsamfsr SUNWsamfsu
```

4. When prompted to define an administrator group, select `yes` or `y` to accept the default (no administrator group), or select no or n if you want to define an administrator group.
   You can reset permissions on certain commands later by using the `set_admin`(1M) command. For more information about this command, see Adding the Administrator Group or the `set_admin`(1M) man page.

5. Examine the SAM-QFS installation log file `/tmp/SAM_install.log`.
   This file should show that the `pkgadd`(1M) command added the `SUNWsamfsr` and `SUNWsamfsu` software packages. Verify that the SAM-QFS `samst` driver is also installed. If all files installed properly, the following message appears:

```
Restarting the sysevent daemon
```

How to Set Up PATH and MANPATH Variables

To enable access to the commands and man pages for the Sun QFS and SAM-QFS commands, you must modify your `PATH` and `MANPATH` environment variables.

1. For users who will need to access the user commands, such as `sls`(1), add `/opt/SUNWsamfs/bin` to the users' `PATH` variables.
2. Edit your system setup files to include the correct paths to commands and man pages.
   - In the Bourne or Korn shells, change the `PATH` and `MANPATH` variables in the `.profile` file and export the variables.
     The following example shows how your `.profile` file might look after editing.

```
PATH=$PATH:/opt/SUNWsamfs/bin:/opt/SUNWsamfs/sbin
MANPATH=$MANPATH:/opt/SUNWsamfs/man
export PATH MANPATH
```

   - In the C shell, edit the `.login` and `.cshrc` files.
     The `path` statement in your `.cshrc` file might look like the following example:

```
set path = ($path /opt/SUNWsamfs/bin /opt/SUNWsamfs/sbin)
```

     The `MANPATH` statement in your `.login` file might look like the following example:

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

Configuring the File System Environment

Each SAM-QFS software environment is unique. The system requirements and hardware differ from site to site. SAM-QFS environments support a wide variety of tape and optical devices, automated libraries, and disk drives. The system administrator at your site must set up the specific configuration for your environment.

The master configuration file, `/etc/opt/SUNWsamfs/mcf`, defines the equipment topology managed by the SAM-QFS software. This file specifies the devices, automated libraries, and file systems included in the environment. You assign each piece of equipment a unique Equipment Identifier in the `mcf` file.

You can edit the `mcf` file in either of two ways:

- By using the SAM-QFS Manager interface to configure archiving and file system devices. When you create a file system using SAM-QFS Manager, it creates an `mcf` file in `/etc/opt/SUNWsamfs/mcf` that contains a line for each device and family set of the file system.

- By directly editing the `mcf` file using a text editor.

The `mcf` file has two kinds of entries:

- File system device entries for disk devices. In the `mcf` file, you organize them into one or more file systems.
- Removable media device entries that you can organize into family sets. The `mcf` file contains information that enables you to identify the drives to be used and associate them with the automated libraries to which they are attached.

For detailed information about `mcf` file structures and contents, see About the Master Configuration File.

Example mcf files are installed in /opt/SUNWsamfs/examples. Several example mcf file configurations are also provided in Examples of mcf Files.

The following sections provide examples and describe activities related to creating and maintaining the mcf file.

How to Create the Master Configuration File (mcf) Manually

- Use vi(1) or another editor to create the /etc/opt/SUNWsamfs/mcf file.
  For detailed information about the contents of the mcf file, see About the Master Configuration File.

  > You can copy an example mcf file from /opt/SUNWsamfs/examples or from the examples in Examples of mcf Files.

  When you create the mcf file, follow these guidelines:

  - Delimit the fields in each line with spaces or tabs.
  - Begin each comment line entered into this file with a pound sign (#).
  - Use a dash (-) to indicate optional fields that are omitted.
    The following example shows the mcf file fields.

    ```
    #
    # Sun  Storage Archive Manager file system configuration
    #
    # Equipment       Equip  Equip Fam   Dev    Additional
    # Identifier      Ord    Type  Set   State  Parameters
    # ----------      -----  ----- ----  -----  ----------
    ```

    The mcf file can contain both comment lines and lines that pertain to a device. The types of lines that can pertain to a device are as follows:

  - Family set parent identifiers and family set devices
  - Family set member devices
  - Stand-alone devices

Identifying Peripherals Using the /var/adm/messages File

When your system boots, a series of messages is written to /var/adm/messages. These messages identify the Sun Solaris hardware path to each of the peripherals on your system. You can use this information to create the mcf file. To display information from the latest system reboot, search backward from the end of the file.

As the following example shows, each SCSI peripheral has three lines. The sixth field, samst2, indicates that these lines are associated with each other.

Example – SCSI Peripheral Lines in the **/var/adm/messages** File

```
# tail -200 /var/adm/messages | more
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@5,8800000/samst@2,0
```

- The first line displays the vendor and product information that the SCSI peripheral reported to the Sun Solaris kernel.
- The second line displays the SCSI bus, SCSI target ID, and LUN of the peripheral.
- The third line displays the peripheral's hardware path. This path is reflected in the /devices directory. Symbolic links (symlinks) to the /devices directory are set up in the /dev/st, /dev/samst, and /dev/rmt directories. Note that the third line wraps to the next line.

Matching the symbolic link to the correct peripheral is the key to configuring a Sun Storage Archive Manager environment. Use the ls(1) command with the -l option in both the /dev/st, /dev/samst and /dev/rmt directories to display the path name of the peripheral.

You might also want to set up the "device down" notification script at this point. The dev_down.sh(1M) man page contains information about setting up this script, which sends email to root when a device is marked down or off. For more information, see the dev_down.sh(1M) man page.

How to Verify the mcf File

- If you created your mcf file manually, use the sam-fsd(1M) command to verify the file.
  If you created your mcf file using SAM-QFS Manager, you do not need to verify its syntax.
  If the mcf file is free of syntax errors, the sam-fsd(1M) output includes information about the file systems, archiving, and other system information. If the mcf file contains syntax or other errors, however, the output is similar to the following example.

```
# sam-fsd
13: /dev/dsk/c1t1d0s0   10      md      samfs1  on      /dev/rdsk/c1t1d0s0
*** Error in line 13: Equipment name '/dev/dsk/c1t1d0s0' already in use by eq 10
72: /dev/rmt/3cbn       45      ug      l1000   on
*** Error in line 72: Equipment name '/dev/rmt/3cbn' already in use by eq 44
2 errors in '/etc/opt/SUNWsamfs/mcf'
sam-fsd: Read mcf /etc/opt/SUNWsamfs/mcf failed.
```

  If the mcf file has errors, return to Setting Up the Environment Configuration and refer to the mcf(4) man page for information about creating this file. You can also refer to Examples of mcf Files.

How to Create an mcf File Using SAM-QFS Manager

Before You Begin

When you configure QFS file systems using the SAM-QFS Manager software, it creates or edits the appropriate configuration files, including the mcf file, on that server. You can use either SAM-QFS Manager or the CLI to further edit these files later.

> Note -
> If you want to use SAM-QFS Manager to configure your archiving environment and you want to include network attached libraries (excluding STK Libraries) in this configuration, you must create your parameters file before you create the mcf file. For information about creating a parameters file, see Creating Parameters Files for Network Attached Automated Libraries.
> You can add a StorageTek ACSLS network library in the SAM-QFS Manager without creating the parameters file. The application automatically generates the parameters file for you when you add the library in the Library Summary Page.

Steps

1. From your web browser, log in to the SAM-QFS Manager as an administrator user.
2. Expand the Getting Started section and choose First Time Configuration.
3. In section 2, click Create a File System.
   The New File System wizard is displayed.
4. Follow the steps for creating a new file system.
   When you have completed this process, the mcf file is created. For more information, see the SAM-QFS Manager online help.

Setting Up Mount Parameters

Use the procedures in this section to specify mount parameters for the file system.

You can specify mount parameters in the following ways:

- Using the mount(1M) command. Mount options specified here override those specified in the /etc/vfstab file and in the samfs.cmd file.
- In the /etc/vfstab file. Mount options specified here override those specified in the samfs.cmd file.
- In the samfs.cmd file.

For a list of available mount options, see the mount_samfs(1M) man page.

Updating the /etc/vfstab File and Creating the Mount Point

This section describes how to edit the /etc/vfstab file.

The following table shows the values you can provide in the fields in the /etc/vfstab file.

Table – **/etc/vfstab** File Fields

Field	Field Title and Content
1	Device to Mount. The name of the file system to be mounted. This value must be the same as the file system's Family Set name specified in the mcf file.
2	Device to fsck(1M). Must be a dash (-) character, which indicates that there are no options. This character prevents the Solaris system from performing an fsck(1M) process on the file system. For more information about this process, see the fsck(1M) or samfsck(1M) man page.
3	Mount Point, for example, /samfs1.
4	File System Type. Must be samfs.
5	fsck(1M) Pass. Must be a dash (-) character, which indicates that there are no options.
6	Mount at Boot. Either yes or no. • Specifying yes in this field indicates that the Sun Storage Archive Manager file system is to be mounted automatically at boot time. • Specifying no in this field indicates that you do not want to mount the file system automatically. For information about the format of these entries, see the mount_samfs(1M) man page.
7	Mount Parameters. A list of comma-separated parameters (with no spaces) that are used in mounting the file system. You can specify mount options on the mount(1M) command, in the /etc/vfstab file, or in a samfs.cmd file. Mount options specified on the mount(1M) command override those specified in the /etc/vfstab file and in the samfs.cmd file. Mount options specified in the /etc/vfstab file override those in the samfs.cmd file. For a list of available mount options, see the mount_samfs(1M) man page.

When you create a file system using SAM-QFS Manager, a default /etc/vfstab file is created. However, mount options specified in SAM-QFS Manager are written to the samfs.cmd file rather than to the /etc/vfstab file. For more information, see Creating and Editing the samfs.cmd File.

To edit the mount options in the /etc/vfstab file, use the following procedure, How to Update the /etc/vfstab File and Create the Mount Point Using a Text Editor.

How to Update the /etc/vfstab File and Create the Mount Point Using a Text Editor

The example in this task assumes that /samfs1 is the mount point of the samfs1 file system.

1. In the /etc/vfstab file, create an entry for each file system.
   The following example shows header fields and entries for a local file system.

```
#DEVICE    DEVICE   MOUNT   FS    FSCK  MOUNT    MOUNT
#TO MOUNT  TO FSCK  POINT   TYPE  PASS  AT BOOT  PARAMETERS
#
samfs1     -        /samfs1 samfs  -    yes      high=80,low=60
```

2. Use the mkdir(1M) command to create the mount point.
   For example:

```
# mkdir /samfs1
```

Creating and Editing the samfs.cmd File

You can create the /etc/opt/SUNWsamfs/samfs.cmd file as the place from which the system reads mount parameters. If you are configuring multiple file systems with multiple mount parameters, consider creating this file.

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

How to Create and Edit the samfs.cmd File Using SAM-QFS Manager

If you specify non-default mount options when creating a file system in SAM-QFS Manager, the `samfs.cmd` file is created or updated with those mount options.

Follow these steps to edit a file system's mount options:

1. On the Managed Hosts page, click the name of the server on which the file system is located.
   The File Systems Summary page is displayed.
2. Select the radio button next to the file system whose mount options you want to edit.
3. From the Operations menu, choose Edit Mount Options.
   The Edit Mount Options page is displayed.
4. Make your edits in the fields.
   For more information about the fields on the Edit Mount Options page, see the File System Manager online help.
5. Click Save.
   The new mount options are written to the `samfs.cmd` file.

How to Create and Edit the samfs.cmd File Using a Text Editor

- Use `vi`(1) or another editor to create the `samfs.cmd` file.
  Create lines in the `samfs.cmd` file to control mounting, performance features, or other aspects of file system management. For more information about the `samfs.cmd` file, see The samfs.cmd File or the `samfs.cmd`(4) man page.

Initializing the Environment

This section tells you how to initialize the environment and the file system, and how to mount the file system.

How to Initialize the Environment

1. Use the `samd`(1M) `config` command to initialize the archiving and file system environment.
   For example:

```
# samd config
```

How to Initialize the File System

This procedure describes how to use the `sammkfs`(1M) command and the Family Set names that you have defined to initialize a file system.

> Note -
> The `sammkfs`(1M) command sets one tuning parameter, the disk allocation unit (DAU). You cannot reset this parameter without reinitializing the file system. For information about how the DAU affects tuning, see File Allocation Methods and the `sammkfs`(1M) man page.

1. Use the `sammkfs`(1M) command to initialize a file system for each Family Set name defined in the `mcf` file.

> Caution -
> Running the `sammkfs`(1M) command creates a new file system. It removes all references to the data currently contained in the partitions associated with the file system in the `/etc/opt/SUNWsamfs/mcf` file.

The following example shows the command to initialize a file system with the Family Set name of samfs1.

```
# sammkfs samfs1
sammkfs: Configuring file system
Building "samfs1" will destroy the contents of devices:
    /dev/dsk/c2t0d0s3
    /dev/dsk/c2t0d0s7
Do you wish to continue? [y/N] *y*
total data kilobytes       = 16777728
total data kilobytes free  = 16777152
#
```

The actual numbers returned vary from file system to file system.

Mounting the File System

The mount(1M) command mounts a file system and reads the /etc/vfstab and samfs.cmd configuration files. For information, see the mount_samfs(1M) man page.

How to Mount the File System Using SAM-QFS Manager

1. On the Managed Hosts page, click the name of the server on which the file system is located.
   The File Systems Summary page is displayed.
2. Select the radio button next to the file system that you want to mount.
3. From the Operations menu, choose Mount.

How to Mount the File System From the Command Line

1. Use the mount(1M) command to mount the file system.
   Specify the file system mount point as the argument. For example:

```
# mount /samfs1
```

> Tip -
> If the file system has not been added to the /etc/vfstab file, use the following form of the mount command:
>
> ```
> #mount -F samfs <fs_name> </mount_point>
> ```

2. Use the mount(1M) command with no arguments to verify the mount.
   This step confirms that the file system is mounted and shows how to set permissions. The following example shows the output from a mount(1M) command issued to verify whether example file system samfs1 is mounted.

```
# mount
_<<< information deleted >>>_
/samfs1 on samfs1 read/write/setuid/intr/largefiles/onerror=panic/dev=8001e3 on Thu Feb  5
11:01:23 2004
_<<< information deleted >>>_
```

3. (Optional) Issue the chmod(1) and chown(1) commands to change the permissions and ownership of the file system's root directory.
   Typically, this step is performed if this is the first time that the file system has been mounted. For example:

```
# chmod 755 /samfs1
# chown root:other /samfs1
```

The page Upgrading SAM and QFS does not exist.

If you want to use the browser user interface to configure and manage your SAM-QFS environment, follow the instructions in this section.

About SAM-QFS Manager

The SAM-QFS Manager is a browser interface tool that enables you to configure, control, protect, and monitor the archiving and file systems in your network from a central location. To access this central location, you can use the web browser on any host in your network.

The goal of the software is to provide a less complex way than command-line interface (CLI) commands to perform the most common archiving and file system tasks.

By default, SAM-QFS Manager is set up to manage the server on which it is installed. It can also be used to manage other servers running Sun Storage Archive Manager software, but those additional servers must first be configured to allow SAM-QFS Manager access. For instructions on how to add additional managed servers, see How to Add an Additional Server for SAM-QFS Manager Access.

Installing SAM-QFS Manager

Perform the tasks in this section to install the SAM-QFS Manager software.

Verifying Requirements for SAM-QFS Manager

You can install the SAM-QFS Manager software in one of the following configurations:

- As a stand-alone management station to manage one or more SAM-QFS Manager hosts
- As additional software on the SAM-QFS Manager host

After the SAM-QFS Manager software is installed, you can invoke SAM-QFS Manager from any machine on the network that is allowed access to its web server.

The host upon which you are configuring SAM-QFS Manager must meet the requirements described in the following sections.

Hardware Requirements

The minimum hardware requirements for the SAM-QFS Manager software are as follows:

- SPARC 400-MHz (or more) CPU or x64 AMD CPU
- 1 gigabyte of memory
- One 20-gigabyte disk
- At least 250 Mbytes free space in `/tmp`
- At least 100 Mbytes free space in `/` (the root partition)
- One 10/100/1000Base-T Ethernet port

Browser Requirements

Ensure that your installation meets the following browser requirements:

- One of the following browsers, at the minimum level indicated, must be used to access the File System Manager software:
  - Netscape™ 7.1 / Mozilla™ 1.7 / Firefox 1.5 on the Solaris OS or Microsoft Windows 98 SE, ME, 2000, or XP operating system
  - Internet Explorer 6.0 on the Microsoft Windows 98 SE, ME, 2000, or XP operating system
- You must enable JavaScript™ technology in your browser. Specific locations differ by browser. For example, in Firefox, go to the Content section of the Preferences panel and confirm that the box next to Enable JavaScript is checked.
- If you are upgrading from an earlier version of the SAM-QFS Manager software, you must clear the browser cache before using SAM-QFS Manager for the first time.

Operating System Requirements

Make sure that the following minimum level of the Solaris OS is installed on the web server:

- Solaris 10 Update 6

Web Software Requirements

The SAM-QFS Manager installation packages include revisions of the following software at the minimum levels indicated:

- Java™ 2 Standard Edition version 1.5.0
- JavaHelp™ 2.0
- Java Studio Enterprise Web Application Framework (JATO) 2.1.2
- Tomcat version 4.0.5

During the installation procedure, you will be asked to answer questions. Based on your answers, the installation software can install the correct revisions for you if the compatible revisions of these software packages are not present.

> Note -
> SAM-QFS Manager is registered in the Sun Java Web Console and can coexist with other applications that use the same console. The Java Web Console uses port 6789. This is an IANA-reserved port, so no application other than Java Web Console should use this port.

How to Install SAM-QFS Manager

Before You Begin

Ensure that you have met the installation requirements described in Verifying Requirements for SAM-QFS Manager.

Steps

1. Log in to the server that you want to use as the SAM-QFS management station.
   You can use the same server on which you installed the `SUNWsamfsr` and `SUNWsamfsu` packages, or you can use a different server on the same network.
2. Become superuser.
3. Go to the directory where the software package release files reside on your server.
4. Execute the `fsmgr_setup` script to start the installation process.
   For example:

```
# fsmgr_setup
```

5. Answer the questions as prompted by the `fsmgr_setup` script.
   During the installation procedure, you are asked questions about your environment.
   The `fsmgr_setup` script automatically installs the following:
   - The `SUNWfsmgrr` package.
   - The `SUNWfsmgru` package.
     The installation script prompts you to specify whether you want to install localized packages.
     After installing the packages, the install software starts the Tomcat Web Server and enables logging.
6. Edit your system setup files to include the correct paths to commands and man pages.
   - In the Bourne or Korn shell, edit the `.profile` file, change the PATH and MANPATH variables, and export the variables.
     The following code example shows how your `.profile` file might look after editing.

```
PATH=$PATH:/opt/SUNWfsmgr/bin
MANPATH=$MANPATH:/opt/SUNWfsmgr/man
export PATH MANPATH
```

   - In the C shell, edit the `.login` and `.cshrc` files.
     When you have finished editing, the `path` statement in your `.cshrc` file might look like the following line:

```
set path = ($path /opt/SUNWfsmgr/bin)
```

The following code example shows how the MANPATH in your .login file might after you have finished editing.

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

7. Use the ps(1) and grep(1) commands to make sure that the rpcbind service is running:

```
# ps -ef | grep rpcbind
```

8. Examine the output from the preceding commands.
   The output should contain a line similar to the following:

```
root    269      1  0    Feb 08 ?        0:06 /usr/sbin/rpcbind
```

   If rpcbind does not appear in the output, type the following command to start the rpcbind service:

```
# /usr/sbin/rpcbind
```

9. (Optional) Start the SAM-QFS Manager (fsmgmtd) daemon.
   If you did not choose to start the SAM-QFS Manager daemon automatically during the installation process, do one of the following:
   - Type the following command to start the SAM-QFS Manager daemon and have it restart automatically every time the daemon process dies. With this configuration, the daemon also automatically restarts at system reboot.

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

   - Type the following command if you want the SAM-QFS Manager daemon to run only once and not automatically restart.

```
# /opt/SUNWsamfs/sbin/fsmadm start
```

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

10. (Optional) Give additional users access to SAM-QFS Manager.
    By default, the root user has privileges to perform all operations available from the SAM-QFS software. You can assign other users full or partial access to SAM-QFS Manager operations.
    To give an additional user access to SAM-QFS Manager, use the useradd command. See How to Create Additional SAM-QFS User Accounts and How to Assign Privilege Levels to SAM-QFS Users for information about adding users and assigning SAM-QFS Manager user privilege levels.

How to Start SAM-QFS Manager

Note -
Before you start SAM-QFS Manager, disable all pop-up blockers.

1. Log in to the server where SAM-QFS Manager is installed, or log in to any system that has network access to it.
2. If you upgraded from a previous version of the software, open the web browser and clear the browser cache.
3. From the web browser, start SAM-QFS Manager.

```
https://<hostname>:6789
```

For hostname, type the name of the host where the SAM-QFS Manager software is installed. If you need to specify a domain name in addition to the host name, specify the hostname in this format: hostname.domainname. Note that this URL begins with `https`, not `http`.
The Sun Java Web Console login page is displayed.

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

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

5. At the Password prompt, type the password.
6. Click Log In.
7. In the Storage section of the Applications page, select SAM-QFS Manager.
   You are now logged in to SAM-QFS Manager.

Configuring SAM-QFS Manager

After SAM-QFS Manager is installed, you can log in to the software using the `root` user name and the password for the management station.

The `root` login grants you full administrator privileges to configure, monitor, control, and reconfigure the devices in your SAM-QFS Manager environment. Only the SAM-QFS Manager administrator should log in using the `root` login. All other users should log in using another user name.

By default, SAM-QFS Manager is set up to manage the server on which it is installed. It can also be used to manage other servers running SAM-QFS Manager software, but those additional servers must first be configured to allow SAM-QFS Manager access. For instructions on adding additional managed servers, see How to Add an Additional Server for SAM-QFS Manager Access.

How to Create Additional SAM-QFS User Accounts

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

If you remove the SAM-QFS Manager software, the removal scripts do not remove any additional accounts that you create manually. You must use one or both of the following procedures to administer any accounts you add manually.

Steps

1. Outside of the browser interface, log in to the SAM-QFS management station as `root`.
2. Use the `useradd` and `passwd` commands to add each user.
   For example, to add a user with account name `bobsmith`, type the following:

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

Each user account that you add in this way has read-only viewing privileges for SAM-QFS Manager functions. To add additional privileges see How to Assign Privilege Levels to SAM-QFS Users.

How to Assign Privilege Levels to SAM-QFS Users

You can assign users full or partial access to SAM-QFS Manager functions.

1. Log in to the SAM-QFS management station as root.
2. To specify full or partial configuration privileges for a user, add the following line to the `/etc/user_attr` file.
   account-name**::::auths=**privilege-level

- account-name is the name of the user's account.
- privilege-level is the level of authorization to assign to the user.
  The following table lists the five levels of privileges you can assign to SAM-QFS Manager users.

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

Example – Assigning Full Privileges to a User

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

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

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

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

How to Create a SAM-QFS Manager Account for Multiple Users

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

1. Use the `useradd` and `passwd` commands to add the account.
   For example, to add a user account called `guest` for multiple users, type the following:

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

2. Use the `roleadd` and `passwd` commands to add the role.
   To create a role called `admin` with special privileges within the `guest` account, type the following:

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

3. Specify the privilege levels in the `/etc/user_attr` file.
   To assign the `admin` role privileges to restore and stage file systems, add the following lines to the `/etc/user_attr` file:

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

   In this example, when a user logs in as `guest`, SAM-QFS Manager prompts the user to select either No Role or Admin. If users know

the Admin role password, they can select Admin, provide the Admin password, and have privileges to restore and stage file systems. All other users must select No Role and have read-only privileges.
Because multiple users with the same privilege level can be logged in to the software concurrently, one user's changes could potentially overwrite another user's previous changes. To prevent this situation, develop policies about who can make changes and how to notify others.

How to Add an Additional Server for SAM-QFS Manager Access

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

1. Outside of the browser interface, use the `telnet` utility to connect to the server you want to add.
   Log in as `root`.
2. Use the `fsmadm`(1M) `add` command to add the SAM-QFS management station to the list of hosts that can remotely administer this server.
   Only hosts that are added to the list through this command can remotely administer the server.
   For example:

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

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

How to Set the SAM-QFS Manager Session Timeout

The Java Web Console framework has a default session timeout of 15 minutes. The SAM-QFS Manager installation program changes the session timeout to 60 minutes. You can change the session timeout to a different value, but to preserve security you should not set it to a value greater than 60 minutes.

1. To change the session timeout value, enter the following command on the SAM-QFS management station:

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

Example – Changing the Session Timeout Value

To change the timeout value to 45 minutes, you would type:

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

Uninstalling the SAM-QFS Manager Software

For instructions on uninstalling the Sun QFS or SAM-QFS packages, see Removing the Existing Software.

How to Uninstall the SAM-QFS Manager Software

1. Log in to the server on which SAM-QFS Manager software is installed.
   This is the host on which you ran the `fsmgr_setup` script at installation time.
2. Become superuser.
3. Issue the following command to remove the SAM-QFS Manager software and all the applications that were installed with it:

```
# /var/sadm/samqfsui/fsmgr_uninstall
```

This script prompts you to confirm removal of the Tomcat Web Server, JRE packages, and information pertaining to administrator and user accounts.

End of Complete (Printable) Sun QFS 5.1 Installation Guide

Sun QFS File System Configuration and Administration Guide

Sun QFS File System Configuration and Administration Guide

Contents

File System Overview

The Sun QFS file system is a configurable file system that presents a standard UNIX file system interface to users.

General File System Configurations

The Sun QFS file system can be used in several different configurations:

- Configured as a stand-alone file system on a single host

- Configured as a shared file system, in which multiple hosts can write to and read from the file system
- Configured as a multireader file system, in which only one host can write to the file system, but multiple host can read from the file system
- Configured with the Sun Storage Archive Manager (SAM-QFS) product to support archiving features
- Configured with the Sun Cluster product for failover and high-availability

Unless otherwise noted, file system information throughout this space applies to archiving and non-archiving configurations.

The Sun QFS file system does not require changes to user programs or to the UNIX kernel.

File System Features

Sun QFS file systems provide the following key features:

- Supports flexible volume management
- Supports paged and direct I/O
- Supports very large files
- Provides fast file system recovery
- Enables flexible metadata storage
- Uses virtual file system `vfs/vnode` interface
- Supports shared file systems

Volume Management

Sun QFS file systems support both striped and round-robin disk access. The master configuration file (`mcf`) and the mount parameters specify the volume management features and enable the file system to recognize the relationships between the devices it controls. This is in contrast to most UNIX file systems, which can address only one device or one portion of a device. Sun QFS file systems do not require additional volume management applications. However, if you want to use mirroring for devices in a Sun QFS environment, you must obtain an additional package, such as a logical volume manager.

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

Support for Paged and Direct I/O

The Sun QFS file system supports two different types of I/O:

- Paged I/O – Also known as cached or buffered I/O. When paged I/O is used, user data is cached in virtual memory pages and the kernel writes the data to disk. The standard Solaris OS interfaces manage paged I/O. This is the default type of I/O for Sun QFS .
- Direct I/O – When direct I/O is used, user data is written directly from user memory to disk. You can specify direct I/O by using the Solaris OS `directio`(3C) function call or the `setfa`(1) command with its `-D` option. By using direct I/O, you can realize substantial performance improvements for large block, sequential, aligned I/O.

High Capacity

The Sun QFS software supports files of up to 2^{63} bytes in length. Such very large files can be striped across many disks or RAID devices, even within a single file system. This is true because Sun QFS file systems use true 64-bit addressing, in contrast to standard UNIX file systems, which are not true 64-bit file systems.

The number of file systems that you can configure is virtually unlimited. The volume manager enables each file system to include up to 252 device partitions, typically disk. Each partition can include up to 16 terabytes of data. This configuration offers virtually unlimited storage capacity.

There is no predefined limit on the number of files in a Sun QFS file system. Because the inode space (which holds information about the files) is dynamically allocated, the maximum number of files is limited only by the amount of disk storage available. The inodes are cataloged in the `.inodes` file under the mount point. The `.inodes` file requires a minimum of 512 bytes of storage per file.

In a Sun QFS file system, the inodes are located on the metadata devices and can be separated from the file data devices. In practice, the size of your metadata (mm) devices limits the number of files in a Sun QFS file system, but you can increase the maximum number of files by adding more metadata devices. The hard limit on the number of files is $2^{32}-1$ files, and the recommended limit is 10^{8} files.

Fast File System Recovery

A key function of a file system is its ability to recover quickly after an unscheduled outage. Standard UNIX file systems require a lengthy file system check (`fsck`(1M)) to repair inconsistencies after a system failure.

A Sun QFS file system often does not require a file system check after a disruption that prevents the file system from being written to disk (using `sync`(1M)). In addition, Sun QFS file systems recover from system failures without using journaling. They accomplish this dynamically by using identification records, serial writes, and error checking for all critical I/O operations. After a system failure, even multiterabyte-sized Sun QFS file systems can be remounted immediately.

Metadata Storage

File systems use metadata to reference file and directory information. Typically, metadata resides on the same device as the file data. However, the Sun QFS file system has the option of separating the file system metadata from the file data by storing them on separate devices. The Sun QFS file system enables you to define one or more separate metadata devices in order to reduce device head movement and rotational latency, improve RAID cache utilization, or mirror metadata without mirroring file data.

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

vnode Interface

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

By using the `vfs/vnode` interface, the file system works with the standard Solaris OS kernel and requires no modifications to the kernel for file management support. Thus, the file system is protected from operating system changes and typically does not require extensive regression testing when the operating system is updated.

The kernel intercepts all requests for files, including those that reside in Sun QFS file systems. If a file is identified as a Sun QFS file, the kernel passes the request to the appropriate file system for handling. Sun QFS file systems are identified as type `samfs` in the `/etc/vfstab` file and through the `mount`(1M) command.

Shared File System Support

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

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

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

Sun QFS shared file systems do not support the following:

- These file types:
  - b– Block special files
  - c– Character special files
  - p– FIFO (named pipe) special files
- Segmented files. You cannot implement a SAM-QFS shared file system in a segmented-file environment.
- Mandatory locks. An EACCES error is returned if the mandatory lock is set. Advisory locks are supported, however. For more information about advisory locks, see the `fcntl`(2) system call.

For more information about shared file systems, see Configuring a Shared File System.

Additional File System Features

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

- Application programming interface (API) routines– API routines enable a program to perform various specialized functions, such as preallocating contiguous disk space or accessing a specific striped group. For more information about these routines, see the `intro_libsam`(3) man page.

- Adjustable disk allocation units (DAUs)– The DAU is the basic unit of online storage. The Sun QFS file system software includes an adjustable DAU, which is useful for tuning file systems with the physical disk storage device and for eliminating the system overhead caused by read-modify-write operations. For information about adjusting the DAU size, see Specifying Disk Allocation Units.
- Support for multiple striped groups– To support multiple RAID devices in a single file system, Sun QFS software supports the definition of striped groups. You can optimize disk block allocation for a striped group, thereby reducing the overhead for updating the on-disk allocation map. Users can assign a file to a striped group either through an API routine or by using the `setfa(1)` command.

File System Design Basics

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

Sun QFS file systems use a linear search method for directory lookups, searching from the beginning of the directory to the end. As the number of files in a directory increases, the search time through the directory also increases. Search times can become excessive when you have directories with thousands of files. These long search times are also evident when you restore a file system. To increase performance and speed up file system dumps and restores, keep the number of files in a directory under 10,000.

The directory name lookup cache (DNLC) feature improves file system performance. This cache stores the directory lookup information for files whose paths are short (30 characters or less), removing the need for directory lookups to be performed on the fly.

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

Inode Files and File Characteristics

The types of files to be stored in a file system affect file system design. An inode is a 512-byte block of information that describes the characteristics of a file or directory. This information is allocated dynamically within the file system.

Inodes are stored in the `.inodes` file located under the file system mount point.

Like a standard Solaris OS inode, a Sun QFS file system inode contains the file's POSIX standard inode times: file access, file modification, and inode changed times. A Sun QFS file system inode includes other times as well, as shown in the following table.

Table – Content of ***.inode*** Files

Time	Incident
`access`	Time the file was last accessed. POSIX standard.
`modification`	Time the file was last modified. POSIX standard.
`changed`	Time the inode information was last changed. POSIX standard.
`attributes`	Time the attributes specific to the file system were last changed. Sun Microsystems extension.
`creation`	Time the file was created. Sun Microsystems extension.
`residence`	Time the file changed from offline to online or vice versa. Sun Microsystems extension.

> Note -
> If the WORM-FS (write once read many) package is installed, the inode also includes a `retention-end` date. See Configuring WORM-FS File Systems for more information.

For more information on viewing inode file information, see Viewing Files and File Attributes.

Specifying Disk Allocation Units

Disk space is allocated in basic units of online disk storage called disk allocation units (DAUs). Whereas sectors, tracks, and cylinders describe the physical disk geometry, the DAU describes the file system geometry. Choosing the appropriate DAU size and stripe size can improve performance and optimize magnetic disk usage. The DAU setting is the minimum amount of contiguous space that is used when a file is allocated.

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

DAU Settings and File System Geometry

Sun QFS file systems use an adjustable DAU. You can configure the DAU to tune the file system to the physical disk storage device. This feature minimizes the system overhead caused by read-modify-write operations and is therefore particularly useful for applications that manipulate very large files. For information about how to control the read-modify-write operation, see Increasing File Transfer Performance for Large Files.

Each file system can have its own unique DAU setting, even if it is one of several mounted file systems active on a server. The possible DAU settings differ depending on the type of file system you are using. The DAU setting is determined through the `sammkfs(1M)` command when the file system is created. It cannot be changed dynamically.

DAU settings work in conjunction with the device and file system definitions specified in the master configuration (`mcf`) file. For details about the `mcf` file, see About the Master Configuration File and the `mcf(4)` man page.

`ms` and `ma` File Systems

Two file allocation schemes are available to you:

- An `ms` file system type - File system data and file system metadata are on the same device
- An `ma` file system type - File system data and file system metadata are on different devices

For a simple Sun QFS file system, such as one on a single partition, the file system is defined in your `mcf` file by an Equipment Type value of `ms`. In the `ms` file system, the only device type allowed is type `md`, and both metadata and file data are written to the `md` devices. By default, the DAU on an `md` device is 64 kilobytes.

A more complex Sun QFS file system installed on multiple partitions is defined as Equipment Type `ma` in your `mcf` file. In an `ma` file system, metadata is written to `mm` devices, and data can be written to `md`, `mr`, or gXXX devices.

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

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

For more information on these device types, see About the Master Configuration File.

Dual and Single Allocation Schemes

The `md` and `mm` devices use a dual allocation scheme, as follows:

- On `md` data devices, the small allocation is 4 kilobytes, and the large allocation is a DAU. The default DAU is 64 kilobytes. You can override this default when the file system is initialized by using the `-a` allocation-unit option to the `sammkfs`(1M) command. The DAU size can be 16, 32, or 64 kilobytes.
- When a file is created on an `md` device, the system allocates the first eight addresses of the file in the small allocation. If more space is needed, the file system uses one or more large allocations (DAUs) to expand the file. As a result, I/O performance improves for large files while minimizing the disk fragmentation that can result from many small files.

> Note -
> When using an `ms` file system, the stripe width should be set to greater than zero to stripe metadata information across the disk. However, you should read and understand Stripe Widths on Data Disks before setting the stripe width and DAU size.

- On `mm` metadata devices, the small allocation is 4 kilobytes, and the large allocation is 16 kilobytes. The dual allocation scheme enables the file system to write metadata to disk more efficiently and helps minimize disk fragmentation.

Depending on the type of file data stored in the file system, a larger DAU size can improve file system performance significantly. For information about tuning file system performance, see Advanced File System Topics.

Only `ma` Sun QFS file systems can include devices that use a single allocation scheme. These file systems consist of separate metadata devices and data devices, as follows:

- The metadata devices can be defined only as Equipment Type `mm`.

- The data devices can be defined as Equipment Type `md`, `mr`, or gXXX. The `md` devices are limited to DAU sizes of 16 kilobytes, 32 kilobytes, or 64 kilobytes.
  The `mr` and gXXX devices follow a single allocation scheme. You can mix `mr` and gXXX devices in a file system, but you cannot mix `md` devices with either `mr` or gXXX devices in a file system. The `mr` and gXXX devices can be set to a minimum DAU allocation of 8 kilobytes for devices in an `ma` file system. This setting is optimal for workloads with the majority of file sizes at or below 8 kilobytes.

The DAU size for file systems that use `mr` and gXXX data devices is configurable. The possible DAU sizes that can be used on data devices depend on the Equipment Type value assigned to each data device in the `mcf` file. The following table shows these DAU sizes.

Table – Equipment Type Values and DAU Sizes

Equipment Type	DAU Sizes
`mr` or gXXX	You can specify different DAU sizes by adjusting the default size in 8-kilobyte increments. The DAU size can be from 8 kilobytes to 65,528 kilobytes (64 megabytes). The default DAU size is 64 kilobytes for `mr` or 256 kilobytes for gXXX.
`md`	This type of device uses a dual allocation scheme. The DAU can be configured to be 16, 32, or 64 kilobytes in length. The default DAU size is 64 kilobytes.An `md` device in an `ma` file system is used to store data only, not metadata. An `md` device in an `ms` file system is used to store both file data and metadata.

> Note -
> If you created your file system using version 3.5 of the software, or built it using the `sammkfs` compatibility mode flag in version 4 of the software, you may be using a version 1 superblock. In the version 1 superblock, `mm` devices do not use the dual allocation scheme, and the allocation for `mm` devices is 16 kilobytes. Only a version 2 superblock enables you to define `md` devices in a Sun QFS file system. To find out whether you are using a version 1 superblock, use the `samfsinfo`(1M) command.

Data Alignment

Data alignment refers to matching the allocation unit of the RAID controller with the allocation unit of the file system. The optimal file system alignment formula is as follows:

allocation-unit = RAID-stripe-width x number-of-data-disks

For example, suppose a RAID-5 unit has nine disks, with one of the nine being the parity disk, making the number of data disks eight. If the RAID stripe width is 64 kilobytes, then the optimal allocation unit is 64 multiplied by 8, which is 512 kilobytes.

Data files are allocated as striped or round-robin through each striped group (gXXX) or data disk (`mr` or `md`) within the same file system.

A mismatched alignment hurts performance because it can cause a read-modify-write operation.

Stripe Widths on Data Disks

Stripe width defaults differ between `ms` and `ma` file systems. The stripe width is specified by the `-o stripe=`n option in the `mount`(1M) command. If the stripe width is set to 0, round-robin allocation is used.

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

Stripe Widths on *ms* File Systems

On `ms` file systems, the stripe width is set at mount time. The following table shows default stripe widths.

Table – *ms* File System Default Stripe Widths

DAU	Default Stripe Width	Amount of Data Written to Disk
16 kilobytes	8 DAUs	128 kilobytes
32 kilobytes	4 DAUs	128 kilobytes
64 kilobytes (default)	2 DAUs	128 kilobytes

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

Note -

- To strip metadata information across the disk in an `ms` file system, set the stripe width to greater than zero.
- If you multiply the number in the first column of the above table by the number in the second column, the resulting number is 128 kilobytes. Sun QFS file systems operate most efficiently if the amount of data being written to disk is at least 128 kilobytes.

Stripe Widths on *ma* File Systems Not Using Striped Groups

On `ma` file systems, the stripe width that is set at mount time depends on whether or not striped groups are configured. A striped group is a collection of devices that are striped as a group. For more information about striped groups, see File Allocation Methods. This section describes stripe widths for Sun QFS file systems that are configured without stripe groups.

If striped groups are not configured, the DAU and stripe width relationships on `ma` file systems are similar to those for `ms` file systems. The difference is that DAUs larger than 64 kilobytes are possible and that the DAU is configurable in 8-kilobyte blocks. The maximum DAU size is 65,528 kilobytes.

By default, if no stripe width is specified, the amount of data written to disk is at or near 128 kilobytes. Sun QFS file systems are most efficient if write operations write at least one whole stripe per I/O request. The following table shows the default stripe widths.

Table – Default Stripe Widths for *ma* File Systems Not Using Striped Groups

DAU	Default Stripe Width	Amount of Data Written to Disk
16 kilobytes	8 DAUs	128 kilobytes
24 kilobytes	5 DAUs	120 kilobytes
32 kilobytes	4 DAUs	128 kilobytes
40 kilobytes	3 DAUs	120 kilobytes
48 kilobytes	2 DAUs	96 kilobytes
56 kilobytes	2 DAUs	112 kilobytes
64 kilobytes (default)	2 DAUs	128 kilobytes
72 kilobytes	1 DAU	72 kilobytes
128 kilobytes	1 DAU	128 kilobytes
> 128 kilobytes	1 DAU	DAU size

Stripe Widths on *ma* File Systems Using Striped Groups

If striped groups are configured for your file system, the minimum amount of space allocated is the DAU multiplied by the number of devices in the striped group. The amount of the allocation can be very large with striped groups.

When striped groups are used, data is written to several disk devices at once, as if they were one device. Allocations on striped groups are equal to the DAU size multiplied by the number of elements in the striped group.

The `-o stripe=n` mount option determines the number of allocations that occur on each stripe group before the allocation moves to a different striped group. If a file system is mounted with `-o stripe=0`, the allocation is always to one striped group.

By default, the setting is `-o stripe=0`, which specifies the round-robin allocation method. The setting can be as low as `-o stripe=0` (which disables striping) or as high as `-o stripe=255`. The system sets `-o stripe=0` if mismatched striped groups are present, in which case a file can reside on only one striped group.

For more information on allocation methods, see File Allocation Methods.

Stripe Widths on Metadata Disks

You can use the `-o mm_stripe=n` option to the `mount_samfs`(1M) command to stripe metadata information on the metadata disk. The default stripe width is `-o mm_stripe=1`, which specifies that the file system write one 16-kilobyte DAU to a metadata disk before switching to the next metadata disk. The small 4-kilobyte DAU is used for metadata disks.

By default, if you have multiple metadata devices, metadata is allocated as specified in the `- o mm_stripe=n` option to the `mount(1M)` command. The setting can be as low as `-o mm_stripe=0`, which disables striping, or as high as `-o mm_stripe=255`.

File Allocation Methods

The Sun QFS software enables you to specify both round-robin and striped allocation methods. The following table shows the default file allocation methods used.

Table – Default Allocation Methods

File System	Metadata	File Data
Standalone file system	Striped	Striped
Shared file system	Striped	Round-robin
Striped groups	Striped	Round-robin

The rest of this section describes allocation in more detail.

Metadata Allocation

Metadata allocation varies according to the type of file system:

- For `ms` file systems, metadata is allocated across the `md` devices.
- For `ma` file systems, metadata is allocated across the `mm` devices. No file data is allocated on the `mm` devices.

Inodes are 512 bytes in length. Directories are initially 4 kilobytes in length. The following table shows how the system allocates metadata.

Table – Metadata Allocation

Metadata Type	Allocation Increments for **ma** File Systems	Allocation Increments for **ms** File Systems
Inodes (`.inodes` file)	16-kilobyte DAU	16-, 32-, or 64-kilobyte DAU
Indirect blocks	16-kilobyte DAU	16-, 32-, or 64-kilobyte DAU
Directories	4-kilobyte blocks and 16-kilobyte DAUs	4 kilobytes, up to a 32- kilobyte total, then DAU size

Round-Robin Allocation

The round-robin allocation method writes one data file at a time to each successive device in the family set. Round-robin allocation is useful for multiple data streams, because in this type of environment aggregate performance can exceed striping performance.

Round-robin disk allocation enables a single file to be written to a logical disk. The next file is written to the next logical disk, and so on. When the number of files written equals the number of devices defined in the family set, the file system starts over again with the first device selected. If a file exceeds the size of the physical device, the first portion of the file is written to the first device, and the remainder of the file is written to the next device with available storage. The size of the file being written determines the I/O size.

To specify round-robin allocation explicitly, enter `stripe=0` in the `/etc/vfstab` file.

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

Figure 1 – Round-Robin Allocation in an ***ms*** File System Using Five Devices

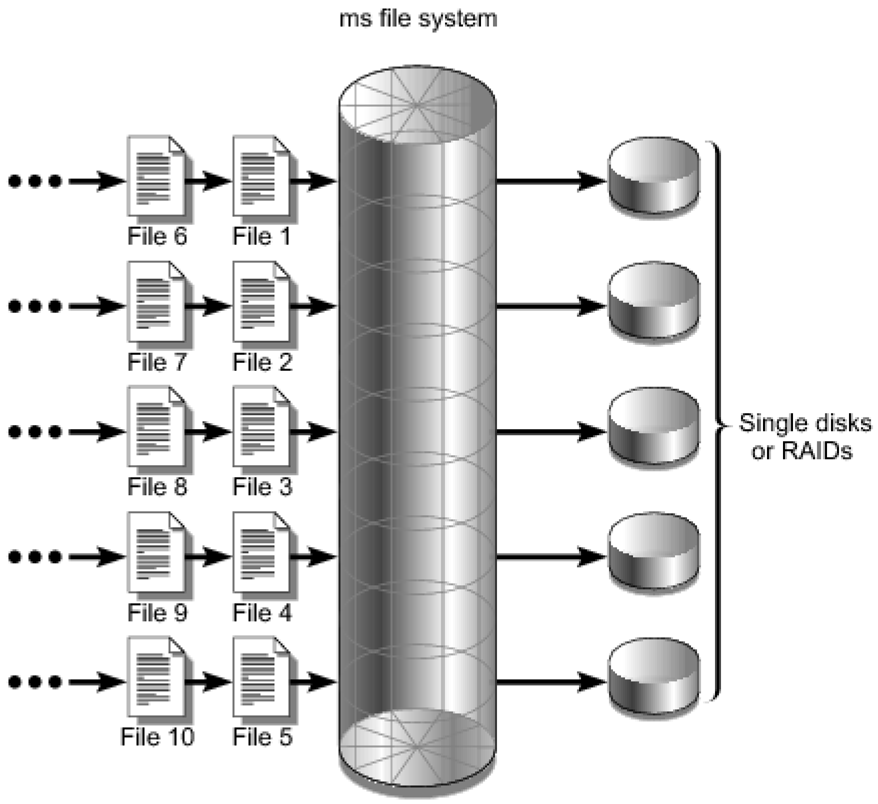
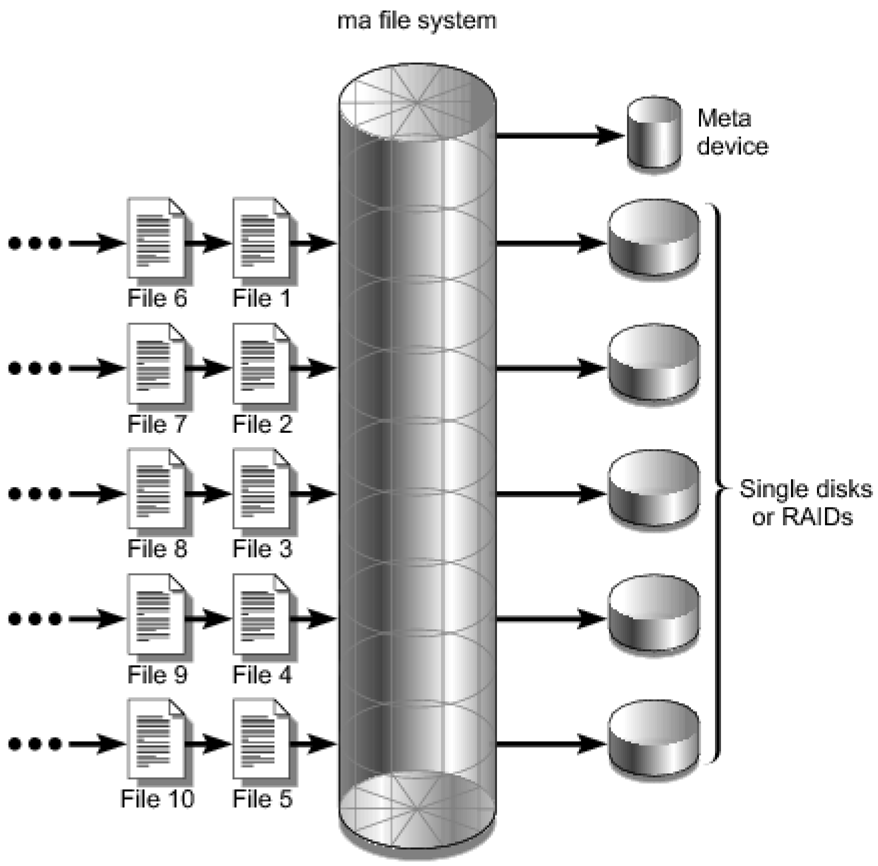


Figure 2 – Round-Robin Allocation in an ***ma*** File System Using Five Devices



Striped Allocation

By default, Sun QFS file systems use a striped allocation method to spread data over all the devices in the file system family set. Striping is a method of concurrently writing files in an interlaced fashion across multiple devices.

Striping is used when performance for one file requires the additive performance of all the devices. A file system that uses striped devices addresses blocks in an interlaced fashion rather than sequentially. Striping generally increases performance because it enables multiple I/O streams to write a file simultaneously across multiple disks. The DAU and the stripe width determine the size of the I/O transmission.

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

When a file system writes a file to an md device, it starts by trying to fit the file into a small DAU, which is 4 kilobytes. If the file does not fit into the first eight small DAUs (32 kilobytes) allocated, the file system writes the remainder of the file into one or more large DAUs.

When a file system writes a file to an `mr` device, it writes first to one DAU, then to another, and so on. The `mr` devices have only one DAU size.

Multiple active files cause significantly more disk head movement with striped allocation than with round-robin allocation. If I/O is to occur to multiple files simultaneously, use round-robin allocation.

The following figures depict ms and ma file systems that use striped allocations. In these figures, DAU `x stripe-width` bytes of the file are written to disk 1. DAU `x stripe-width` bytes of the file are written to disk 2 and so on. The order of the stripe is first-in-first-out for the files. Striping spreads the I/O load over all the disks.

Figure 3 – Striping in an ***ms*** File System Using Five Devices

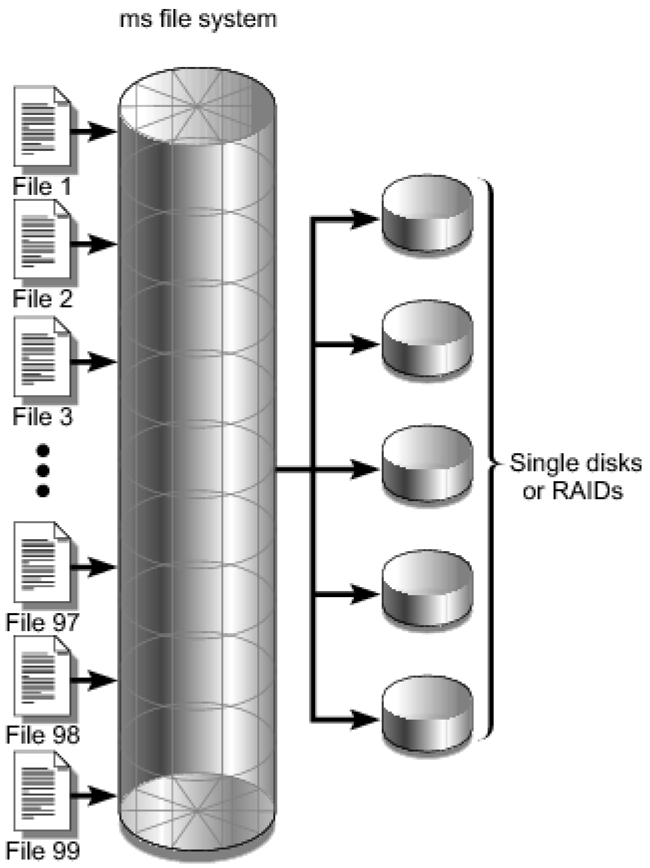
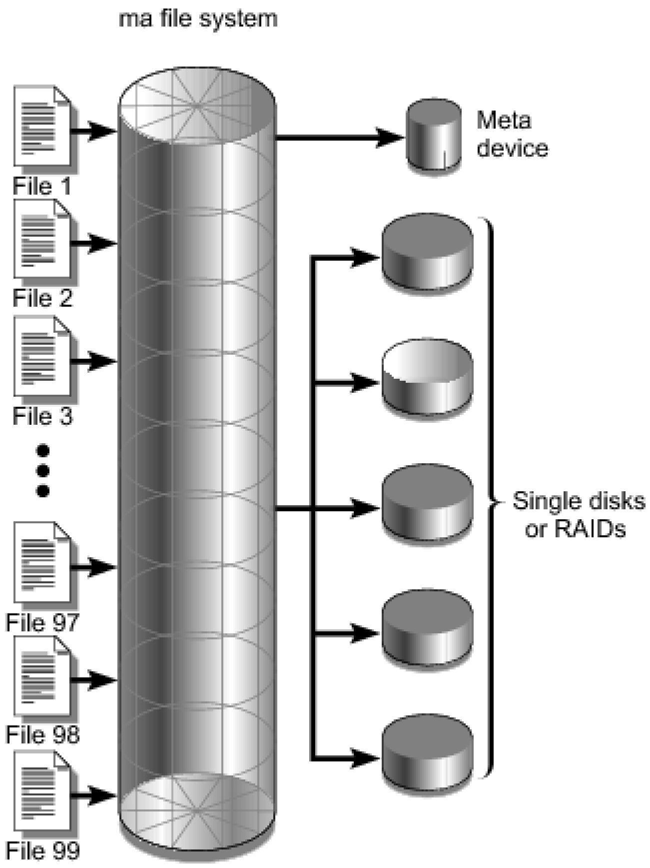


Figure 4 – Striping in an ***ma*** File System Using Five Devices



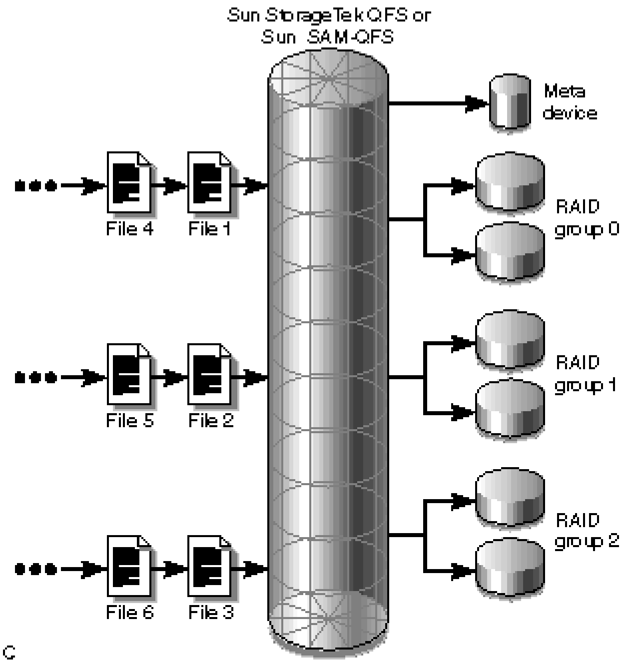
Striped Groups

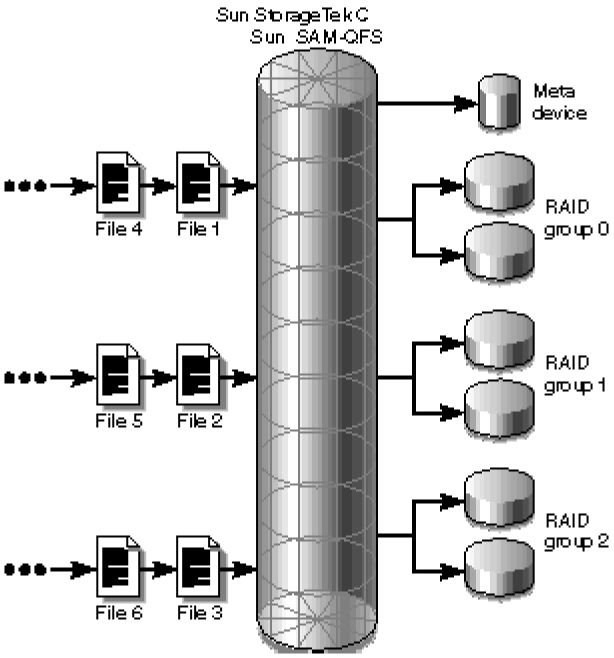
A striped group is a Sun QFS allocation method designed for file systems that have extremely large I/O requirements and terabytes of disk cache. A striped group enables you to designate an Equipment Type value that accounts for multiple physical disks. Multiple striped group Equipment Type entries can make up a single Sun QFS file system. Striped groups save bitmap space and system update time for very large RAID configurations.

A striped group is a collection of devices within a Sun QFS file system. Defined in the `mcf` file as gXXX devices, striped groups enable one file to be written to and read from two or more devices. You can specify up to 128 striped groups within a file system.

Figure 5 – Sun QFS Round-Robin Striped Groups

The following figure depicts an `ma` file system using striped groups and a round-robin allocation. In this figure, files written to the `qfs1` file system are allocated round-robin among the defined striped groups `g0`, `g1`, and `g2`. Each group consists of two physical RAID devices.





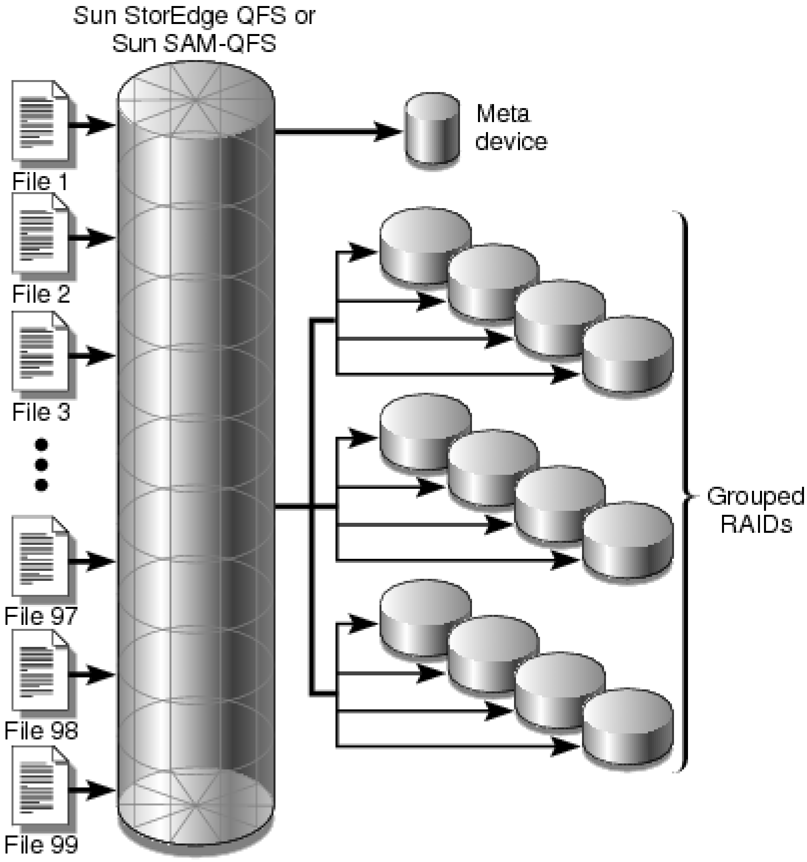
For the configuration in the above figure, the mount point option in `/etc/vfstab` is set to `stripe=0`. The following example shows the `mcf` file that declares these striped groups.

Example – ***mcf*** File Showing Striped Groups

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

Figure 6 – Sun QFS Striped Group Allocation

The following figure depicts a Sun QFS ma file system using striped groups and striped allocation. Files written to the `qfs1` file system are striped through groups g0, g1, and g2. Each group includes four physical RAID devices. The mount point option in /etc/vfstab is set to stripe=1 or greater.



Mismatched Striped Groups

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

> Note -
> In a file system that contains mismatched striped groups, a single file cannot span multiple stripe groups. If the stripe group on which the file resides fills, it cannot be extended. If mismatched stripe groups are present, use the setfa(1) command's -g option to direct files into the desired group. For more information, see the setfa(1) man page.
> To determine how full a stripe group is, use the samu(1M) operator utility, and access the m display to display the status of mass storage.

The following example shows how a file system can be set up with mismatched striped groups to store different types of files.

Example of a Mismatched Striped Group

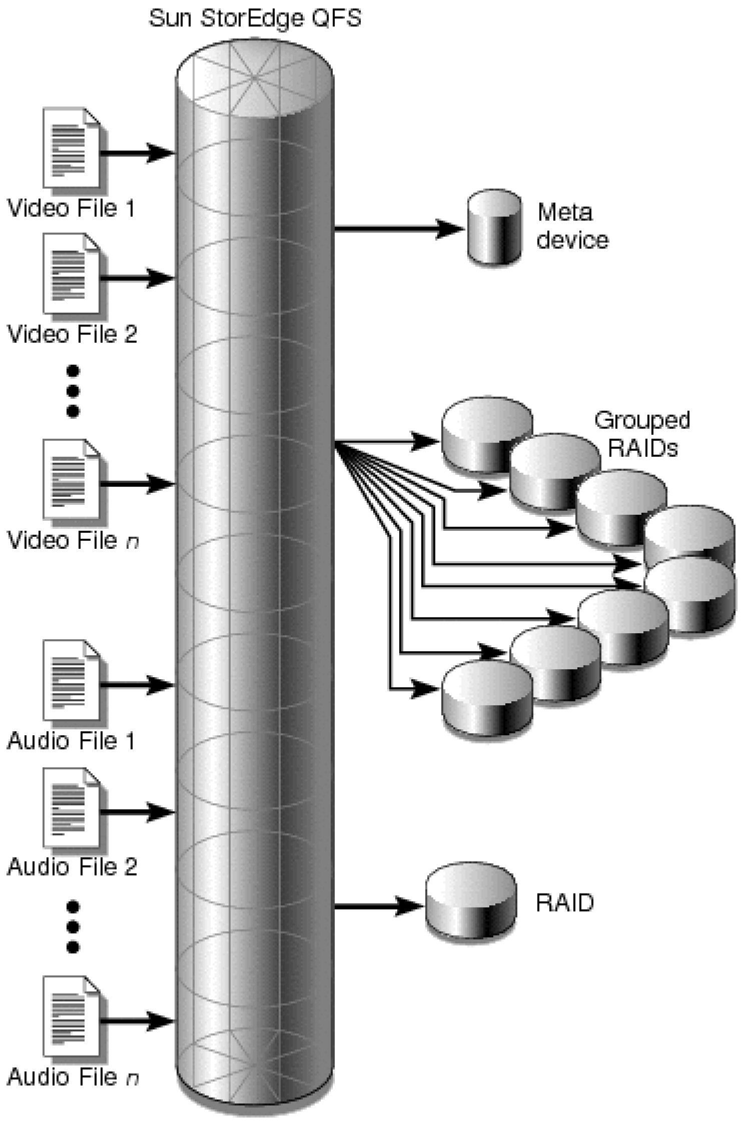
Suppose that you need to create a file system at your site that contains both video and audio data.

Video files are quite large and require greater performance than audio files. You want to store them in a file system with a large striped group, because striped groups maximize performance for very large files.

Audio files are smaller and require lower performance than video files. You want to store them in a small striped group. One file system can support both video and audio files.

The following figure depicts the file system needed. It is an ma file system that uses mismatched striped groups in a striped allocation.

Figure 7 – Sun QFS File System Using Mismatched Striped Groups in a Striped Allocation



The following table shows the characteristics of this sample file system.

Table – Sample File System Characteristics

Characteristics	Notes
File system name	`avfs`
Number of stripe groups	Two. The video file group is `g0`. The audio file group is `g1`.
Stripe width	0
DAU	128 kilobytes
Number of disks for `g0`	Eight
Minimum block size for `g0`	Eight disks x 128-kilobyte DAU = 1024 kilobytesThis is the amount of data written in one block write. Each disk receives 128 kilobytes of data, so the total amount written to all disks at one time is 1024 kilobytes.
Number of disks for `g1`	One

Minimum block size for g1	One disk x 128-kilobyte DAU = 128 kilobytes

Add the following line to the /etc/vfstab file so that the environment recognizes the avfs file system:

```
avfs    -    /avfs    samfs    -    no    stripe=0
```

Note that in the /etc/vfstab file, stripe=0 is used to specify a round-robin file system. This is used because a value greater than 0 is not supported for mismatched striped groups.

The following example shows the mcf file for file system avfs.

Example – The ***mcf*** File for File System ***avfs***

```
# Equipment Eq Eq Fam Dev Additional
# Identifier Ord Type Set State Parameters
#
avfs 100 ma avfs
/dev/dsk/c00t1d0s6 101 mm avfs -
#
/dev/dsk/c01t0d0s6 102 g0 avfs -
/dev/dsk/c02t0d0s6 103 g0 avfs -
/dev/dsk/c03t0d0s6 104 g0 avfs -
/dev/dsk/c04t0d0s6 105 g0 avfs -
/dev/dsk/c05t0d0s6 106 g0 avfs -
/dev/dsk/c06t0d0s6 107 g0 avfs -
/dev/dsk/c07t0d0s6 108 g0 avfs -
/dev/dsk/c08t0d0s6 109 g0 avfs -
#
/dev/dsk/c09t1d0s6 110 g1 avfs -
```

When the mcf file for this file system is ready, you can enter the sammkfs(1M) and mount(1M) commands shown in the following example to create and mount the avfs file system.

Example – Commands to Create and Mount File System ***avfs***

```
# sammkfs -a 128 avfs
# mount avfs
```

After the file system is mounted, you can use the commands shown in the following example to create two directories for the two types of files.

Example – Commands to Create Directories in File System ***avfs***

```
# cd /avfs
# mkdir video
# mkdir audio
```

Once the directories are created, you can use the setfa(1) commands shown in the following example to assign the large striped group to video and the small striped group to audio. Files created in these directories are allocated on their respective striped groups because attributes are inherited.

Example – Commands to Set File Attributes

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

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

Per-Logical Unit Number (LUN) Allocation Control

If necessary, you can disable allocation to a specific Sun QFS data partition by using a nalloc command, which prohibits any future allocation to that device. The feature is currently only usable on data partitions, not on metadata partitions.

Allocation to a partition can be restarted by either an alloc or on command.

The allocation state of a partition (allocflag) is persistent across boots.

The nalloc and alloc commands are available in the samu interface, and the samu on command also sets allocation to on. The samu screens display the nalloc state for partitions that have been disabled. The samtrace and samfsinfo output also include the allocation state.

For more information about the samu interface, see Using the samu Operator Utility.

About the Master Configuration File

The master configuration file /etc/opt/SUNWsamfs/mcf describes all devices that are under the control of, or used by, the Sun QFS or SAM-QFS software. When you create this ASCII file at system configuration time, you declare attributes for each device, and you group the devices in each file system into family sets.

The mcf file contains the information that these file systems need in order to identify and organize RAID and disk devices into file systems. It also contains entries for each automated library or device included in a file system. A sample mcf file is located in /opt/SUNWsamfs/examples/mcf.

Basic **mcf** File Structure

An mcf file consists of lines of specification code divided into six columns, or fields, as shown in the following example.

```
Equipment   Equipment  Equipment  Family  Device  Additional
Identifier  Number     Type       Set     State   Parameters
```

Follow these rules when entering data in the mcf file:

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

For more information about writing the mcf file, see the mcf(4) man page.

You can also use SAM-QFS Manager to automatically create an mcf file. For information about installing SAM-QFS Manager, see Installing SAM-QFS Manager. For information about using SAM-QFS Manager, see its online help.

mcf File Fields

- The Equipment Identifier Field (Required)
- The Equipment Number Field (Required)
- The Equipment Type Field (Required)
- The Family Set Field (Required)
- The Device State Field (Optional)
- The Additional Parameters Field (Optional)

The Equipment Identifier Field (Required)

The Equipment Identifier field identifies the physical file system device or removable media device. If this field contains the name of a file system, it is limited to 31 characters. For all other content, this field is limited to 127 characters.

Use the Equipment Identifier field to provide the following information:

Information	Identifer Length	Description
The file system name	31 characters	The file system name must be identical to the name in the Family Set field, and the subsequent lines in the `mcf` file must define all the disks or devices included in the file system. More than one file system can be declared in an `mcf` file. Typically, the first data line in an `mcf` file declares the first file system, and subsequent lines specify the devices included in the file system. Other file systems declared in the `mcf` file can be preceded by a blank comment line for readability. Note - File system names must start with an alphabetic character and can contain only alphabetic characters, numeric characters, or underscore (_) characters.
The nodev keyword	127 characters	The keyword `nodev` indicates that the system on which the `mcf` file resides is being used as a client host in a shared file system on a Solaris host. This keyword can appear in this field only as the Equipment Identifier for one or more metadata devices that reside on the metadata server. For more information about creating an `mcf` file for the members of a shared file system, see Configuring a Shared File System. Note - Do not use this keyword if your file system is in a Sun Cluster environment.
A disk partition or slice description	127 characters	A `/dev/` entry in this field identifies a disk partition or slice.
An automated library or optical drive description	127 characters	A `/dev/samst` entry identifies an automated library or optical drive. If you are configuring a network attached automated library, see Creating Parameters Files for Network Attached Automated Libraries.
A tape drive description	127 characters	This entry can be in one of two forms: • A `/dev/rmt` entry. • A path to a symbolic link that points to the same file to which the `/dev/rmt` link points. If you specify a tape drive in this manner, be sure to create the link before mounting the file system.

The Equipment Number Field (Required)

For each row in the `mcf` file, the Equipment Number (`eq`) field must contain a unique numeric identifier for the file system component or device being defined. This number must be an integer between 1 and 65534, inclusive.

> Tip -
> Use low numbers to keep the internal software tables small.

The Equipment Type Field (Required)

The required Equipment Type field provides information that the software uses to determine how to interact with a particular device. Enter the two- or three-character mnemonic for the device type.

Some equipment can use the generic equipment types of `od` (optical disk), `tp` (tape), and `rb` (robot). For a file system, the following table describes specific Equipment Type codes:

Table – Equipment Type Field

Equipment Type	Description
ms	Defines a file system that stores both data and metadata on the same device (an md device).
ma	Defines a file system that stores metadata on a separate device (an mm device). The data in an ma file system can be stored on md , mr, or g_XXX devices.
md	Defines a striped or round-robin device that uses dual allocation for storing file data. See also Dual and Single Allocation Schemes.
mm	Defines a metadata device for storing inode and other metadata information. You can specify multiple metadata devices. Metadata (including inodes, directories, allocation maps, and so on) on ma file systems is located on metadata devices, separated from the file data devices. By default, metadata is allocated using round-robin allocation if you have multiple metadata devices.
mr	Defines a round-robin or striped data device that uses single allocation for storing file data. See also Dual and Single Allocation Schemes.
gXXX	Defines a striped data device. Striped groups start with the letter g followed by a number. The number must be an integer between 0 and 127, inclusive; for example, g12.All members in a striped group must be the same type and size. Different striped groups within one file system are not required to have the same number of members. md, mr, and gXXX devices cannot be mixed in one file system. Data can be striped (if all groups contain the same number of devices) or round-robin between groups. The default is round-robin.

Besides the file system equipment types, other codes are used to identify automated libraries and other devices. For more information about specific equipment types, see the mcf(4) man page.

The Family Set Field (Required)

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

Family set names must start with an alphabetic character and can contain only alphabetic characters, numeric characters, or underscore (_) characters. The family set name cannot be longer than 31 characters.

The family set name can be one of the following:

Family Set	Description
File system name	All disk devices in the file system must use the same file system name in this field. The software uses the family set name to group devices together as a file system. It physically records the family set name on all of the devices in the file system when the sammkfs(1M) command is issued. You can change this name by using the -F and -R options together in the samfsck(1M) command. For more information about the sammkfs(1M) command, see the sammkfs(1M) man page. For more information about the samfsck(1M) command, see the samfsck(1M) man page.
Automated library identifier	The library and all its associated drive devices must use the same identifier.
-	The dash character (-) indicates a standalone removable media device.

You can create a comment that is associated with a specific family set by inserting the identifier #family-set-name: just before the first device in that family set. Any comments that are added between that comment line and the last device in the family set will be associated with that family set. If the family set is later deleted through the SAM-QFS Manager software, any related comments will also be deleted from the mcf file.

The Device State Field (Optional)

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

The Additional Parameters Field (Optional)

For an automated library device, the Additional Parameters field is optional and can be left blank. By default, library catalog files are written to /var/opt/SUNWsamfs/catalog/family-set-name. Use this field if you want to specify an alternative path to the library catalog file.

For a shared file system, this field must contain the keyword `shared`.

For other entries, enter a dash (-) or leave this field blank.

Examples of mcf Files

Each file system configuration is unique. System requirements and actual hardware differ from site to site. The following code examples show sample `mcf` files. More complete examples that include information on how you can duplicate the configuration are available in mcf File Examples.

Example – Example *mcf* File Showing Striped Groups

This example shows an `mcf` file for a Sun QFS file system with two striped groups.

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

Example – Example *mcf* File Showing Three File Systems

This example shows an `mcf` file with three file systems.

```
# SAM-QFS file system configuration example
#
# Equipment  Eq  Eq   Fam. Dev. Additional
# Identifier Ord Type Set State Parameters
#----------- --- -- ------ ------ -----------------
qfs1 10 ma qfs1 -
/dev/dsk/c1t13d0s6 11 mm qfs1 -
/dev/dsk/c1t12d0s6 12 mr qfs1 -
#
qfs2 20 ma qfs2 -
/dev/dsk/c1t5d0s6 21 mm qfs2 -
/dev/dsk/c5t1d0s6 22 mr qfs2 -
#
qfs3 30 ma qfs3 -
/dev/dsk/c7t1d0s3 31 mm qfs3 -
/dev/dsk/c6t1d0s6 32 mr qfs3 -
/dev/dsk/c6t1d0s3 33 mr qfs3 -
/dev/dsk/c5t1d0s3 34 mr qfs3 -
```

Example – Example *mcf* File Showing a File System and a Library

This example shows an `mcf` file with one archiving file system that uses md devices. This `mcf` file also defines a tape library.

```
# Equipment Eq Eq Fam. Dev. Additional
# Identifier Ord Type Set State Parameters
#----------- --- -- ------ ------ ----------
samfs1 10 ma samfs1 -
/dev/dsk/c1t2d0s6 11 mm samfs1 -
/dev/dsk/c1t3d0s6 12 md samfs1 -
/dev/dsk/c1t4d0s6 13 md samfs1 -
/dev/dsk/c1t5d0s6 14 md samfs1 -
# scalar 1000 and 12 AIT tape drives
/dev/samst/c5t0u0 30 rb robot1 -
/dev/rmt/4cbn 101 tp robot1 on
/dev/rmt/5cbn 102 tp robot1 on
/dev/rmt/6cbn 103 tp robot1 on
/dev/rmt/7cbn 104 tp robot1 off
/dev/rmt/10cbn 105 tp robot1 on
/dev/rmt/11cbn 106 tp robot1 on
/dev/rmt/3cbn 107 tp robot1 on
/dev/rmt/2cbn 108 tp robot1 on
/dev/rmt/1cbn 109 tp robot1 on
/dev/rmt/0cbn 110 tp robot1 on
/dev/rmt/9cbn 111 tp robot1 on
/dev/rmt/8cbn 112 tp robot1 on
```

Interactions Among File Settings, Options, and Directives

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

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

For information about the various ways to specify mount options, see Setting Up Mount Parameters.

`mcf` File Examples

The master configuration file, `/etc/opt/SUNWsamfs/mcf`, defines the topology of the equipment managed by the file system. This file specifies the devices and file systems included in the environment and contains information that enables you to identify the disk slices to be used and to organize them into file systems.

This section provides some specific examples of `mcf` files for various types of file systems.

Configuration Examples for Local File Systems

Use the configuration examples in this section for configuring the `mcf` file for a file system to be installed on a single Solaris OS host.

For `mcf` examples that you can use in a Sun Cluster environment, see Configuration Examples for Highly Available File Systems.

Simple File System Configuration Example

This example shows how to configure two file systems using a server that has a SCSI-attached Sun StorageTek Multipack desktop array.

You can use the `format`(1M) command to determine how the disks are partitioned. The following example shows the `format`(1M) command output.

> Note -
> Only the last lines of `format`(1M) output are shown.

Example – ***format***(1M) Command Output for Configuration Example 1

```
# format < /dev/null
Searching for disks...done

AVAILABLE DISK SELECTIONS:
                0. c0t10d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
                    /sbus@3,0/SUNW,fas@3,8800000/sd@a,0
                1. c0t11d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
                    /sbus@3,0/SUNW,fas@3,8800000/sd@b,0
                2. c6t2d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
                    /pci@7,4000/SUNW,isptwo@3/sd@2,0
                3. c6t3d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
                    /pci@7,4000/SUNW,isptwo@3/sd@3,0
                4. c6t4d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
                        /pci@7,4000/SUNW,isptwo@3/sd@4,0
                5. c6t5d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
                    /pci@7,4000/SUNW,isptwo@3/sd@5,0
                6. c8t2d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
                    /pci@b,4000/SUNW,isptwo@3/sd@2,0
                7. c8t3d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
                    /pci@b,4000/SUNW,isptwo@3/sd@3,0
                8. c8t4d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
                    /pci@b,4000/SUNW,isptwo@3/sd@4,0
                9. c8t5d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
                    /pci@b,4000/SUNW,isptwo@3/sd@5,0
Specify disk (enter its number):

# format /dev/rdsk/c6t2d0s2
.
.
.
Part Tag Flag Cylinders Size Blocks
        0 unassigned wm 0 0 (0/0/0) 0
        1 unassigned wm 0 0 (0/0/0) 0
        2 backup wu 0 - 4923 8.43GB (4924/0/0) 17682084
        3 unassigned wm 0 0 (0/0/0) 0
        4 unassigned wm 0 - 1229 2.11GB (1230/0/0) 4416930
        5 unassigned wm 1230 - 2459 2.11GB (1230/0/0) 4416930
        6 unassigned wm 2460 - 3689 2.11GB (1230/0/0) 4416930
        7 unassigned wm 3690 - 4919 2.11GB (1230/0/0) 4416930
```

How to Configure the System for Simple Example

Begin writing the `mcf` file for this configuration example by defining the file system and its disk partitions, as follows:

1. Write the `mcf` file.
2. Make an `ma` entry for the first file system (`qfs1`).
3. Using the information from the output of the `format` command, make an `mm` entry listing the partitions that constitute the metadata for the `qfs1` file system.
4. Using the information from the output of the `format` command, make a series of `mr` entries listing the partitions that constitute the file data for the `qfs1` file system.
5. Make similar entries for the second file system (`qfs2`).
   The finished `mcf` file defines the following two file systems:
   - The `qfs1` file system, which is created on slice 4 of the following disks: `c8t2d0` (metadata), `c6t2d0` (file data), and `c6t3d0` (file data).
   - The `qfs2` file system, which is created on slice 5 of the following disks: `c8t2d0` (metadata), `c6t2d0` (file data), and `c6t3d0` (file data).
     The following code example shows the resulting `mcf` file.

```
# cat /etc/opt/SUNWsamfs/mcf
#
# Equipment         Eq   Eq     Family   Device   Additional
# Identifier        Ord  Type    Set     State    Parameters
#-----------       ---  ----   ------   ------   ----------
#
*qfs1 10 ma qfs1 on*
/dev/dsk/c8t2d0s4    11    mm   qfs1       on
/dev/dsk/c6t2d0s4    12    mr   qfs1       on
/dev/dsk/c6t3d0s4    13    mr   qfs1       on
#
*qfs2 20 ma qfs2 on*
*/dev/dsk/c8t2d0s5 21 mm qfs2 on*
*/dev/dsk/c6t2d0s5 22 mr qfs2 on*
*/dev/dsk/c6t3d0s5 23 mr qfs2 on*
```

6. Modify the `/etc/vfstab` file.
   Make entries in the `/etc/vfstab` file for the `qfs1` and `qfs2` file systems that you defined in the `mcf` file. The last two lines in the code example below show entries for these new file systems.
   For a description of the fields in the `/etc/vfstab` file, see Table 3-2.

```
# cat /etc/vfstab
# device            device                     file           mount
# to                to                   mount  system   fsck  at    mount
# mount             fsck                 point  type     pass  boot  params
# -----             ----                 -----  ----     ----  ----  ------
fd                  -                    /dev/fd  fd       -     no    -
/proc               -                    /proc    proc     -     no    -
/dev/dsk/c0t10d0s1  -                    -        swap     -     no    -
/dev/dsk/c0t10d0s0  /dev/rdsk/c0t10d0s0  /        ufs      1     no    logging
swap                -                    /tmp     tmpfs    -     yes   -
*qfs1 - /qfs1 samfs - yes stripe=1*
*qfs2 - /qfs2 samfs - yes stripe=1*
```

Round-Robin Configuration Example

This example illustrates the configuration of a file system (called `qfs3`) that uses round-robin allocation on four disk drives.

This example assumes the following:

- The metadata device is a single partition (`s1`) used on controller 8, disk 4.
- The data devices consist of four disks attached to controller 6. Each disk is on a separate target (1–4).

How to Configure the System for Round-Robin Allocation

This example introduces the round-robin data layout. For more information about data layout, see the Sun StorageTek QFS File System Configuration and Administration Guide.

1. Write the `mcf` file as described in Configuration Example 1.
   The following code example shows the `mcf` file for this round-robin disk configuration.

```
# cat /etc/opt/SUNWsamfs/mcf
#
# Equipment          Eq    Eq     Family   Device   Additional
# Identifier         Ord   Type    Set     State    Parameters
#-----------        ---   ----   ------   ------   ----------
#
*qfs3 10 ma qfs3 on*
/dev/dsk/c8t4d0s4    11    mm   qfs3       on
/dev/dsk/c6t2d0s4    12    mr   qfs3       on
/dev/dsk/c6t3d0s4    13    mr   qfs3       on
/dev/dsk/c6t4d0s4    14    mr   qfs3       on
/dev/dsk/c6t5d0s4    15    mr   qfs3       on
```

2. Modify the `/etc/vfstab` file.
   Edit the `/etc/vfstab` file to explicitly set round-robin allocation on the file system by specifying `stripe=0` in the `mount params` field. The following code example shows `stripe=0` for the `qfs3` file system.
   For a description of the fields in the `/etc/vfstab` file, see Table 3-2.

```
# cat /etc/vfstab
#device             device                       file          mount
#to                 to                  mount    system  fsck  at    mount
#mount              fsck                point    type    pass  boot  params
#-----              ----                -----    ----    ----  ----  ------
fd                  -                   /dev/fd  fd      -     no    -
/proc               -                   /proc    proc    -     no    -
/dev/dsk/c0t10d0s1  -                   -        swap    -     no    -
/dev/dsk/c0t10d0s0  /dev/rdsk/c0t10d0s0 /        ufs     1     no    logging
swap                -                   /tmp     tmpfs   -     yes   -
*qfs3 - /qfs3 samfs - yes stripe=0*
```

3. Initialize the file system by using the `sammkfs`(1M) command.
   The default disk allocation unit (DAU) is 64 kilobytes, but the following example sets the DAU size to 128 kilobytes:

```
# sammkfs -a 128 qfs3
```

Local Striping Configuration Example

This example illustrates the configuration of a file system (called `qfs4`) that stripes file data to four disk drives. This example assumes the following:

- The metadata device is a single partition (`s6`) used on controller 0, logical unit number (LUN) 0.
- The data devices consist of four disks attached to controller 6. Each disk is on a separate target (2–5).

How to Configure the System for Local Striping

1. Write the `mcf` file as shown in Configuration Example 1.
   The following code example shows the `mcf` file for this striped disk configuration.

```
# Equipment          Eq    Eq     Family   Device   Additional
# Identifier         Ord   Type    Set     State    Parameters
#-----------        ---   ----   ------   ------   ----------
#
*qfs4 40 ma qfs4 on*
/dev/dsk/c8t4d0s4    41    mm   qfs4       on
/dev/dsk/c6t2d0s4    42    mr   qfs4       on
/dev/dsk/c6t3d0s4    43    mr   qfs4       on
/dev/dsk/c6t4d0s4    44    mr   qfs4       on
/dev/dsk/c6t5d0s4    45    mr   qfs4       on
```

2. Modify the /etc/vfstab file.
Set the stripe width by using the stripe= option. The following code example shows the /etc/vfstab file with a mount parameter of stripe=1 set for the qfs4 file system.
For a description of the fields in the /etc/vfstab file, see Table 3-2.

```
# cat /etc/vfstab
#
#device              device                     file          mount
#to                  to                  mount   system fsck  at    mount
#mount               fsck                point   type   pass  boot  params
#-----               ----                -----   -----  ----  ----  ------
fd                   -                   /dev/fd  fd     -     no    -
/proc                -                   /proc   proc   -     no    -
/dev/dsk/c0t10d0s1   -                   -       swap   -     no    -
/dev/dsk/c0t10d0s0   /dev/rdsk/c0t10d0s0 /       ufs    1     no    logging
swap                 -                   /tmp    tmpfs  -     yes   -
*qfs4 - /qfs4 samfs - yes stripe=1*
```

The stripe=1 specification stripes file data across all four of the mr data disks with a stripe width of one DAU. The DAU is the allocation unit you set when you use the sammkfs(1M) command to initialize the file system.

3. Initialize the Sun StorageTek QFS file system by using the sammkfs(1M) command.
The following example sets the DAU size to 128 kilobytes:

```
# sammkfs -a 128 qfs4
```

With this striped disk configuration, any file written to this file system is striped across all of the devices in increments of 128 kilobytes. Files less than the aggregate stripe width times the number of devices still use 128 kilobytes of disk space. Files larger than 128 kilobytes have space allocated for them as needed in total space increments of 128 kilobytes.

Striped Group Configuration Example

Striped groups enable you to build RAID-0 devices of separate disk devices. With striped groups, however, there is only one DAU per striped group. This method of writing large, effective DAUs across RAID devices saves system update time and supports high-speed sequential I/O. Striped groups are useful for writing very large files to groups of disk devices.

> Note -
> A DAU is the minimum disk space allocated. The minimum disk space allocated in a striped group is as follows:
> allocation-unit x number of disks in the group
> Writing a single byte of data consumes a DAU on every member of the striped group. Make sure that you understand the effects of using striped groups with your file system.

The devices within a striped group must be the same size. It is not possible to increase the size of a striped group. You can add additional striped groups to the file system, however.

This example illustrates the configuration of a file system (called qfs5) that separates the metadata onto a low-latency disk. The mcf file defines two striped groups on four drives. This example assumes the following:

- The metadata device is a single partition (s5) used on controller 8, disk 4.
- The data devices consist of four disks (two groups of two identical disks) attached to controller 6. Each disk is on a separate target (2–5).

How to Configure the System for Striped Groups

1. Write the mcf file as shown in Configuration Example 1.
The following code example shows a sample mcf file for a striped group configuration.

```
# cat /etc/opt/SUNWsamfs/mcf
#
# Equipment         Eq   Eq     Family   Device   Additional
# Identifier        Ord  Type    Set     State    Parameters
#-----------        ---  ----   ------   ------   ----------
#
*qfs5 50 ma qfs5 on*
/dev/dsk/c8t4d0s5    51    mm   qfs5       on
/dev/dsk/c6t2d0s5    52    g0   qfs5       on
/dev/dsk/c6t3d0s5    53    g0   qfs5       on
/dev/dsk/c6t4d0s5    54    g1   qfs5       on
/dev/dsk/c6t5d0s5    55    g1   qfs5       on
```

2. Modify the `/etc/vfstab` file.
   Set the stripe width by using the `stripe=` option. The following code example shows the `/etc/vfstab` file with a mount parameter of `stripe=0`, which specifies round-robin allocation between striped group `g0` and striped group `g1`.
   For a description of the fields in the `/etc/vfstab` file, see “Fields in the /etc/vfstab File” on page 35.

```
# cat /etc/vfstab
#device             device                       file          mount
#to                 to                   mount   system  fsck  at    mount
#mount              fsck                 point   type    pass  boot  params
#-----              ----                 -----   ----    ----  ----  ------
fd                  -                    /dev/fd fd      -     no    -
/proc               -                    /proc   proc    -     no    -
/dev/dsk/c0t10d0s1  -                    -       swap    -     no    -
/dev/dsk/c0t10d0s0  /dev/rdsk/c0t10d0s0  /       ufs     1     no    logging
swap                -                    /tmp    tmpfs   -     yes   -
*qfs5 - /qfs5 samfs - yes stripe=0*
```

3. Initialize the file system by using the `sammkfs`(1M) command.
   The{{– a}} option is not used with striped groups because the DAU is equal to the size of an allocation, or the size, of each group.

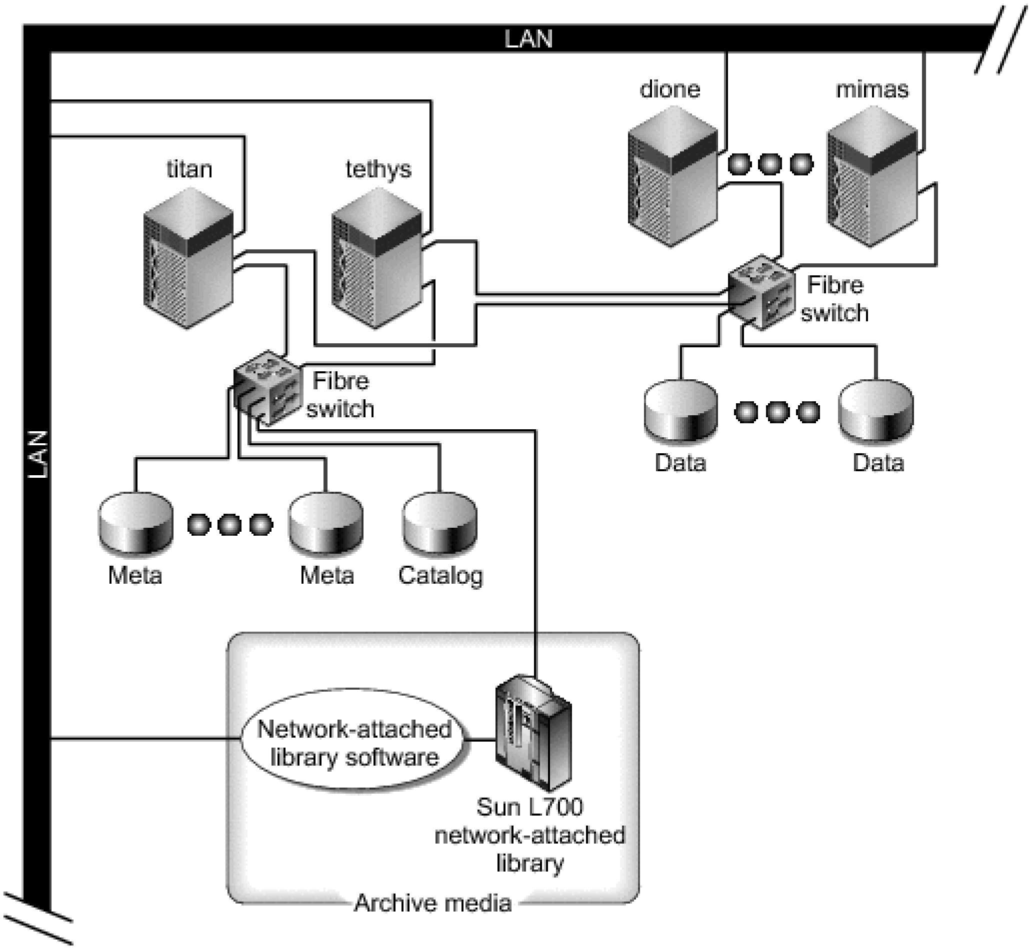
```
# sammkfs qfs5
```

In this example, there are two striped groups, `g0` and `g1`. With `stripe=0` in `/etc/vfstab`, devices 12 and 13 are striped; devices 14 and 15 are striped; and files are round-robined around the two striped groups. A striped group is treated as a bound entity. After you configure a stripe group, you cannot change it without issuing another `sammkfs`(1M) command.

Configuration Example for a Shared File System on a Solaris OS Platform

This figure illustrates a shared file system configuration in an archiving environment.

Figure – Shared File System Configuration



The above figure shows four network attached hosts: `titan`, `tethys`, `dione`, and `mimas`. The `tethys`, `dione`, and `mimas` hosts are clients, and `titan` is the current metadata server. The `tethys` host is a potential metadata server.

The archive media consist of a network attached library and tape drives that are fibre-attached to `titan` and `tethys`. In addition, the archive media catalog resides in a file system that is mounted on the current metadata server, `titan`.

Metadata travels to and from the clients to the metadata server over the network. The metadata server makes all modifications to the name space, thereby keeping the metadata consistent. The metadata server also provides the locking capability, the block allocation, and the block deallocation.

Several metadata disks are connected to `titan` and `tethys` and can be accessed only by the potential metadata servers. If `titan` were unavailable, you could change the metadata server to `tethys`, and the library, tape drives, and catalog could be accessed by `tethys` as part of the Sun StorageTek QFS shared file system. The data disks are connected to all four hosts by a Fibre Channel (FC) connection.

How to Configure the Shared File System

1. Issue the `format`(1M) command and examine its output.
   Make sure that the metadata disk partitions configured for the shared file system mount point are connected to the potential metadata servers. Also make sure that the data disk partitions configured for the shared file system are connected to the potential metadata servers and to all the client hosts in this file system.
   If your host supports multipath I/O drivers, individual devices shown in the output of the `format`(1M) command might display multiple controllers. These correspond to the multiple paths to the actual devices.
   The following code example shows the `format`(1M) command output for `titan`. There is one metadata disk on controller 2, and there are three data disks on controller 3.

```
titan<28>format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c1t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w2100002037e9c296,0
       1. c2t2100002037E2C5DAd0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w2100002037e9c296,0
       2. c2t50020F23000065EEd0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w50020f23000065ee,0
       3. c3t50020F2300005D22d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@8,600000/SUNW,qlc@1/fp@0,0/ssd@w50020f2300005d22,0
       4. c3t50020F2300006099d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@8,600000/SUNW,qlc@1/fp@0,0/ssd@w50020f2300006099,0
       5. c3t50020F230000651Cd0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@8,600000/SUNW,qlc@1/fp@0,0/ssd@w50020f230000651c,0
```

The following code example shows the `format`(1M) command output for `tethys`. There is one metadata disk on controller 2, and there are four data disks on controller 7.

```
tethys<1>format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c0t1d0 <IBM-DNES-318350Y-SA60 cyl 11112 alt 2 hd 10 sec 320>
          /pci@1f,4000/scsi@3/sd@1,0
       1. c2t2100002037E9C296d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w2100002037e9c296,0
       2. c2t50020F23000065EEd0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@4/ssd@w50020f23000065ee,0
       3. c7t50020F2300005D22d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@5/ssd@w50020f2300005d22,0
       4. c7t50020F2300006099d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@5/ssd@w50020f2300006099,0
       5. c7t50020F230000651Cd0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@5/ssd@w50020f230000651c,0
```

Note the following in above code example:

- The data disks on `titan`'s controller 3 are the same disks as on `tethys`'s controller 7. You can verify this by looking at the World Wide Name, which is the last component in the device name. For `titan`'s number 3 disk, the World Wide Name is `50020f2300005d22`. This is the same name as number 3 on controller 7 on `tethys`.
- For `titan`'s metadata disk, the World Wide Name is `50020F23000065EE`. This is the same metadata disk as `tethys`'s controller 2, target 0.
  The following code example shows the `format`(1M) command output for `mimas`. This shows three data disks on controller 1 and no metadata disks.

```
mimas&lt;9>format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
          /pci@1f,4000/scsi@3/sd@0,0
       1. c1t50020F2300005D22d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@4/fp@0,0/ssd@w50020f2300005d22,0
       2. c1t50020F2300006099d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@4/fp@0,0/ssd@w50020f2300006099,0
       3. c1t50020F230000651Cd0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@4/fp@0,0/ssd@w50020f230000651c,0
```

As shown in the above code examples, the data disks on `titan`'s controller 3 are the same disks as those on `mimas`'s controller 1. You can verify this by looking at the World Wide Name, which is the last component in the device name.

> Note -
> All the data disk partitions must be connected and accessible from all the hosts that share this file system. All the disk partitions, for both data and metadata, must be connected and accessible to all potential metadata servers. You can use the `format`(1M) command to verify these connections.
> For some storage devices, it is possible that the `format`(1M) command's output does not present unique worldwide Names. If you find that this is the case, see the `libdevid`(3LIB) man page for information about finding such devices on different hosts.

2. Use `vi`(1) or another editor to create the `mcf` file on the metadata server.
   The only difference between the `mcf` file of a shared file system and an unshared file system is the presence of the `shared` keyword in the Additional Parameters field of the file system name line for a shared file system.

   > Note -
   > If file systems are already operational on the shared file system's metadata server or on any of the client host systems, select a Family Set name and select equipment numbers that do not conflict with existing Family Set names or equipment numbers on any host that will be included in the shared file system.

   The following code example shows a fragment of the `mcf` file for `titan` that defines several disks for use in the shared file system. It shows the `shared` keyword in the Additional Parameters field on the file system name line.

```
# Equipment                       Eq   Eq    Family   Dev   Addl
# Identifier                      Ord  Type  Set      Stat  Params
------------                      ---  ----  ------   ----  ------
sharefs1                          10   ma    sharefs1 on    shared
/dev/dsk/c2t50020F23000065EEd0s6 11   mm    sharefs1 on
/dev/dsk/c3t50020F2300005D22d0s6 12   mr    sharefs1 on
/dev/dsk/c3t50020F2300006099d0s6 13   mr    sharefs1 on
/dev/dsk/c3t50020F230000651Cd0s6 14   mr    sharefs1 on
```

Configuration Examples for Highly Available File Systems

The Sun Cluster software moves a highly available file system from a failing node to a viable node in the event of a node failure.

Each node in the Sun Cluster environment that can host this file system must have an `mcf` file. During the file system configuration process, you copy `mcf` file lines from the metadata server's `mcf` file to other nodes in the Sun Cluster environment. For more information, see Editing `mcf` Files on Other Hosts.

How to Create an mcf File for a Highly Available File System

The procedure for creating an `mcf` file for a highly available file system is as follows:

1. Make an ma entry for the file system.
2. Make an mm entry listing the partitions that constitute the metadata for the `qfs1` file system.
3. Make a series of `mr`, gXXX, or md entries listing the partitions that constitute the file data for the `qfs1` file system.
   You can use the `scdidadm`(1M) command to determine the partitions to use.
   Example 1. The following code example shows an `mcf` file entry for a highly available file system that uses raw devices.

```
#Equipment            Eq   Eq     Family   Additional
#Identifier           Ord  Type   Set      Parameters
#-------------------- ---  ----   ------   ----------
qfs1                   1   ma    qfs1     on
/dev/global/dsk/d4s0  11   mm    qfs1
/dev/global/dsk/d5s0  12   mr    qfs1
/dev/global/dsk/d6s0  13   mr    qfs1
/dev/global/dsk/d7s0  14   mr    qfs1
```

Example 2. The following code example shows an `mcf` file entry for a highly available file system that uses Solaris Volume Manager metadevices. In this example, the Solaris Volume Manager metaset in use is named `red`.

```
#Equipment            Eq   Eq     Family   Additional
#Identifier           Ord  Type   Set      Parameters
#-------------------- ---  ----   ------   ----------
qfs1                   1   ma    qfs1     on
/dev/md/red/dsk/d0s0  11   mm    qfs1
/dev/md/red/dsk/d1s0  12   mr    qfs1
```

Example 3. The following code example shows an `mcf` file entry for a highly available file system that uses VxVm devices.

```
#Equipment            Eq   Eq     Family   Additional
#Identifier           Ord  Type   Set      Parameters
#-------------------- ---  ----   ------   ----------
qfs1                   1   ma    qfs1     on
/dev/vx/dsk/oradg/m1  11   mm    qfs1
/dev/vx/dsk/oradg/m2  12   mr    qfs1
```

Configuration Example for a Shared File System on a Sun Cluster Platform

In this example, `ash` and `elm` are nodes in a Sun Cluster environment. Host `ash` is the metadata server. The keyword `shared` in this example's `mcf` file indicates to the system that this is a shared file system. This example builds upon Example – Verifying Devices and Device Redundancy.

How to Create an mcf File for a Shared File System in a Sun Cluster Environment

You must create the `mcf` file on the node that you want to designate as the metadata server. The procedure for creating an `mcf` file for a shared file system in a Sun Cluster environment is as follows:

1. Use the `scdidadm`(1M) `-L` command to obtain information about the devices included in the Sun Cluster environment.
   The `scdidadm`(1M) command administers the device identifier (DID) devices. The `-L` option lists all the DID device paths, including those on all nodes in the Sun Cluster environment.
   The following code example uses Sun StorageTek T3 arrays in a RAID-5 configuration. The output shows that you can use devices 4 through 9 for configuring the disk cache for a shared file system.

```
ash# scdidadm -L
1        ash:/dev/rdsk/c0t6d0        /dev/did/rdsk/d1
2        ash:/dev/rdsk/c1t1d0        /dev/did/rdsk/d2
3        ash:/dev/rdsk/c1t0d0        /dev/did/rdsk/d3
4        elm:/dev/rdsk/c6t50020F2300004921d1 /dev/did/rdsk/d4
4        ash:/dev/rdsk/c5t50020F2300004921d1 /dev/did/rdsk/d4
5        elm:/dev/rdsk/c6t50020F2300004921d0 /dev/did/rdsk/d5
5        ash:/dev/rdsk/c5t50020F2300004921d0 /dev/did/rdsk/d5
6        elm:/dev/rdsk/c6t50020F23000049CBd1 /dev/did/rdsk/d6
6        ash:/dev/rdsk/c5t50020F23000049CBd1 /dev/did/rdsk/d6
7        elm:/dev/rdsk/c6t50020F23000049CBd0 /dev/did/rdsk/d7
7        ash:/dev/rdsk/c5t50020F23000049CBd0 /dev/did/rdsk/d7
8        elm:/dev/rdsk/c6t50020F23000055A8d0 /dev/did/rdsk/d8
8        ash:/dev/rdsk/c5t50020F23000055A8d0 /dev/did/rdsk/d8
9        elm:/dev/rdsk/c6t50020F23000078F1d0 /dev/did/rdsk/d9
9        ash:/dev/rdsk/c5t50020F23000078F1d0 /dev/did/rdsk/d9
10       elm:/dev/rdsk/c0t6d0        /dev/did/rdsk/d10
11       elm:/dev/rdsk/c1t1d0        /dev/did/rdsk/d11
12       elm:/dev/rdsk/c1t0d0        /dev/did/rdsk/d12
```

2. Using the output from the `scdidadm`(1M) `-L` command, use the `format`(1M) command to display the information for the devices in the Sun Cluster environment. The following code example shows the `format` command output from all the `/dev/did` devices. You will need this information when you build the `mcf` file.

```
ash# format /dev/did/rdsk/d4s2
selecting /dev/did/rdsk/d4s2

Primary label contents:

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 64 sec 32>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =   64
nsect       =   32
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264       16.86GB    (17265/0/0) 35358720
  1        usr    wm   17265 - 34529       16.86GB    (17265/0/0) 35358720
  2     backup    wu       0 - 34529       33.72GB    (34530/0/0) 70717440
  3 unassigned    wu       0                0         (0/0/0)            0
  4 unassigned    wu       0                0         (0/0/0)            0
  5 unassigned    wu       0                0         (0/0/0)            0
  6 unassigned    wu       0                0         (0/0/0)            0
  7 unassigned    wu       0                0         (0/0/0)            0

ash# format /dev/did/rdsk/d5s2
selecting /dev/did/rdsk/d5s2

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 192 sec 64>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =  192
nsect       =   64
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264      101.16GB    (17265/0/0) 212152320
  1        usr    wm   17265 - 34529      101.16GB    (17265/0/0) 212152320
  2     backup    wu       0 - 34529      202.32GB    (34530/0/0) 424304640
  3 unassigned    wu       0                0         (0/0/0)             0
  4 unassigned    wu       0                0         (0/0/0)             0
  5 unassigned    wu       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7 unassigned    wu       0                0         (0/0/0)             0

ash# format /dev/did/rdsk/d6s2
selecting /dev/did/rdsk/d6s2
```

```
Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 64 sec 32>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =   64
nsect       =   32
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264       16.86GB    (17265/0/0) 35358720
  1        usr    wm   17265 - 34529       16.86GB    (17265/0/0) 35358720
  2     backup    wu       0 - 34529       33.72GB    (34530/0/0) 70717440
  3 unassigned    wu       0                0         (0/0/0)            0
  4 unassigned    wu       0                0         (0/0/0)            0
  5 unassigned    wu       0                0         (0/0/0)            0
  6 unassigned    wu       0                0         (0/0/0)            0
  7 unassigned    wu       0                0         (0/0/0)            0


ash# format /dev/did/rdsk/d7s2
selecting /dev/did/rdsk/d7s2

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 192 sec 64>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =  192
nsect       =   64
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264      101.16GB    (17265/0/0) 212152320
  1        usr    wm   17265 - 34529      101.16GB    (17265/0/0) 212152320
  2     backup    wu       0 - 34529      202.32GB    (34530/0/0) 424304640
  3 unassigned    wu       0                0         (0/0/0)             0
  4 unassigned    wu       0                0         (0/0/0)             0
  5 unassigned    wu       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7 unassigned    wu       0                0         (0/0/0)             0

ash# format /dev/did/rdsk/d8s2
selecting /dev/did/rdsk/d8s2

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 128 sec 128>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =  128
nsect       =  128
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264      134.88GB    (17265/0/0) 282869760
  1        usr    wm   17265 - 34529      134.88GB    (17265/0/0) 282869760
  2     backup    wm       0 - 34529      269.77GB    (34530/0/0) 565739520
  3 unassigned    wu       0                0         (0/0/0)             0
  4 unassigned    wu       0                0         (0/0/0)             0
  5 unassigned    wu       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7 unassigned    wu       0                0         (0/0/0)             0


ash# format /dev/did/rdsk/d9s2
selecting /dev/did/rdsk/d9s2

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 128 sec 128>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =  128
nsect       =  128
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264      134.88GB    (17265/0/0) 282869760
  1        usr    wm   17265 - 34529      134.88GB    (17265/0/0) 282869760
  2     backup    wu       0 - 34529      269.77GB    (34530/0/0) 565739520
  3 unassigned    wu       0                0         (0/0/0)             0
```

```
4 unassigned    wu       0                0         (0/0/0)             0
5 unassigned    wu       0                0         (0/0/0)             0
```

```
 6 unassigned    wu       0                0         (0/0/0)            0
 7 unassigned    wu       0                0         (0/0/0)            0
```

The format(1M) command reveals the space available on a device, but it does not reveal whether a disk is mirrored or striped. The above code example's format(1M) output reveals the following information that is used during creation of the mcf file shown in the next code example:

- Output for devices d4s0 and d6s0 shows 16.86 Gbytes each. These devices are assigned equipment numbers 501 and Equipment Number 502, respectively, in the mcf file. They are the appropriate size to use for metadata slices.
- Output for devices d8s0 and d9s0 shows 134.88 Gbytes each. These devices are assigned equipment number 503 and Equipment Number 504, respectively, in the mcf file. They are the are the appropriate size to be used for storing data.

3. Make an ma entry for the file system.
   In this line entry, include the shared keyword in the Additional Parameters field.
4. Make an mm entry listing the partitions that constitute the metadata for the qfs1 file system.
   Put the file system's mm devices on mirrored (RAID-1) disks. The mm devices should constitute about 10% of the space allocated for the entire file system.
5. Make a series of mr entries listing the partitions that constitute the file data for the qfs1 file system.
   The following code example shows the resulting mcf file.

```
#Equipment            Eq   Eq     Family   Additional
#Identifier           Ord  Type   Set      Parameters
#-------------------- ---  ----   ------   ----------
#
# Family Set sqfs1 (shared FS for SunCluster)
#
sqfs1                500   ma    sqfs1    shared
sqfs1                500   ma    sqfs1    shared
/dev/did/dsk/d4s0    501   mm    sqfs1    -
/dev/did/dsk/d6s0    502   mm    sqfs1    -
/dev/did/dsk/d8s0    503   mr    sqfs1    -
/dev/did/dsk/d9s0    504   mr    sqfs1    -
```

Configuring the File System

This section provides information about configuring an archiving or standalone file system. For information about configuring a shared file system, see Configuring a Shared File System.

Function of the **mcf** File

The master configuration file (mcf), located in /etc/opt/SUNWsamfs/mcf, describes all devices that the Sun QFS or SAM-QFS software controls or uses. When you create this ASCII file at system configuration time, you declare attributes for each device, and you group the devices in each file system into family sets.

The mcf file contains the information that the file systems need in order to identify and organize RAID and disk devices into file systems. It also contains entries for each automated library or device included in a file system. A sample mcf file is located in /opt/SUNWsamfs/examples/mcf.

For more information about the mcf file, see About the Master Configuration File and the mcf(4) man page.

Initializing a File System

To create a new file system or to replace an old or damaged file system, use the sammkfs(1M) command to initialize the file system. The sammkfs(1M) command constructs and initializes file systems.

> Note -
> Beginning with Sun QFS 5.0, the `sammkfs` command creates a version 2A file system that has new features, but is not compatible with previous releases. Use the `sammkfs -P` format to create a version 2 file system that is backwards compatible with previous releases. Use the `sammkfs -a` allocation-unit option to specify the DAU setting.
>
> The following `samfsinfo`(1M) command output shows that the `samfs1` file system is using a version 2 superblock.
>
> ```
> # samfsinfo samfs1
> name:     samfs1       version:        2
> time:     Wed Feb 21 13:32:18 1996
> count:    1
> capacity:      001240a0          DAU:         16
> space:         000d8ea0
> ord  eq   capacity      space   device
>   0  10   001240a0   000d8ea0   /dev/dsk/c1t1d0s0
> ```
>
> For more information about features that require a version 2A superblock, or about using the `sammkfs`(1M) command to create the version 2 superblock, see Support for Version 2A Superblock.

The following example shows the `sammkfs`(1M) command in its simplest form, with the file system name as its only argument. This command builds a version 2A superblock for a stand-alone Sun QFS or SAM-QFS file system.

```
# sammkfs samqfs1
```

For more information about the `sammkfs`(1M) command, its options, and the implications of the version 2 and 2A superblocks, see the `sammkfs`(1M) man page . For information about using the `sammkfs`(1M) command to initialize a shared Sun QFS file system, see Configuring a Shared File System.

Configuration Examples

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

Note that all sample configurations could have automated libraries and other removable media devices defined as well, essentially extending the file system beyond the size of the disk cache. Removable media device configurations are shown in only one example. For information about configuring removable media devices, see Configuring Storage Devices for Archiving.

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

How to Create a Round-Robin Disk Configuration

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

This procedure assumes the following:

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

Steps

1. Use an editor to create the `mcf` file, as shown in the following code example.

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

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

```
# mkdir /qfs
```

3. Use the `sammkfs`(1M) command to initialize the file system.
   The following example uses the default 64-kilobyte DAU:

```
# sammkfs qfs1
```

4. Use an editor to modify the `/etc/vfstab` file.
   The Sun QFS file system with `mr` data devices uses striped allocation as a default, so you must set `stripe=0` for round-robin allocation. To explicitly set round-robin on the file system, set `stripe=0`, as follows:

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

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

```
# mount /qfs
```

How to Create a Striped Disk Configuration

In this sample configuration, file data is striped to four data partitions by default.

This procedure assumes the following:

- The metadata device is a single partition (s6) used on controller 0, LUN 1. Metadata is written to equipment number 11 only.
- The data devices consist of four disks attached to four controllers. Each disk is on a separate controller.

Steps

1. Use an editor to create the `mcf` file, as shown in the following code example.

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

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

```
# mkdir /qfs
```

3. Use the sammkfs(1M) command to initialize the file system.
   The default DAU is 64 kilobytes, but the following example sets the DAU size to 128 kilobytes:

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

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

4. Use an editor to modify the /etc/vfstab file.
   The Sun QFS file system uses striped allocation by default. This example sets the stripe width as stripe=1, which is the default. The following setting stripes data across all four of the mr devices with a stripe width of 1 DAU:

```
qfs1    -    /qfs    samfs    -    yes    stripe=1
```

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

```
# mount /qfs
```

How to Create a Striped Group Configuration

Striped groups enable you to group RAID devices together for very large files. A DAU is represented by one bit in the bitmap. If the striped group has n devices, n multiplied by the DAU is the minimum allocation. Only one bit in the bitmap is used to represent n x DAU.

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

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

> Caution -
> Writing a single byte of data fills the entire minimum disk space allocated in a striped group. Striped groups are used for very specific applications. Make sure that you understand the effects of using striped groups with your file system.

Files with lengths less than the aggregate stripe width times the number of devices (in this example, files less than 128 kilobytes x 4 disks = 512 kilobytes in length) still use 512 kilobytes of disk space. Files larger than 512 kilobytes have space allocated for them as needed in total space increments of 512 kilobytes.

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

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

This procedure assumes the following:

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

Steps

1. Use an editor to create the mcf file, as shown in the following code example.

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

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

```
# mkdir /qfs
```

3. Use the `sammkfs(1M)` command to initialize the file system.
   The following example sets the DAU size to 128 kilobytes:

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

4. Use an editor to modify the `/etc/vfstab` file.
   The following example uses the default setting of `stripe=0`, which essentially specifies a round-robin allocation from striped group `g0` to striped group `g1`:

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

   This `/etc/vfstab` file sets the stripe width using the `stripe=` option. In this example, there are two striped groups, `g0` and `g1`. With the `stripe=0` specification, files are written round-robin around the two striped groups.

   Note -
   To change the configuration of the striped group after it is created, you must issue another `sammkfs(1M)` command.

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

```
# mount /qfs
```

Configuring a Shared File System

For information about configuring a shared file system in a Sun Cluster environment, see Configuring Sun QFS File Systems With Solaris Cluster.

Using Shared QFS With NFS

Note -
If you are using NFS v4, you must disable delegations before you can use shared QFS.

Starting with SAM-QFS 5.0 on Solaris 10 OS, the Service Management Facility (SMF) is used to manage the mounting of the file system at boot time. If your file system uses NFS, the exact sequence in which you configure NFS and shared QFS is important. If you do not follow the steps below, either the shared QFS mount or the NFS share will succeed, and the other will fail.

How to Configure Shared QFS With NFS

1. Export the existing NFS configuration to a file.
   The following example exports the configuration into a file `/var/tmp/server.xml`.

```
svccfg export /network/nfs/server > /var/tmp/server.xml
```

2. Edit the file that you just exported.
   For example:

```
vi /var/tmp/server.xml
```

3. After the local file system dependency listed in the file, add a dependency to mount QFS file systems before you NFS share them.
   For example:

```
  <!--
    Must have QFS filesystems mounted before sharing them
  -->
     <dependency name='qfs'
     grouping='require_all'
     restart_on='error'
     type='service'>
     <service_fmri value='svc:/network/qfs/shared-mount:default'/>
     </dependency>
```

4. Validate the changes that you made to the file.

```
svccfg validate /var/tmp/server.xml
```

5. Disable NFS.

```
svcadm disable nfs/server
```

6. Delete the existing NFS server configuration.

```
svccfg delete nfs/server
```

7. Import the file that you edited in Step 2 into SMF.

```
svccfg import /var/tmp/server.xml
```

8. Enable NFS.
   NFS uses the updated file and reads the QFS dependency information.

```
svcadm enable nfs/server
```

9. Confirm that the dependency is applied.

```
svcs -d svc:/network/nfs/server:default
```

Mounting and Unmounting Shared File Systems

When you mount or unmount a shared file system, the order in which you mount or unmount the metadata server and the clients is important.

For failover purposes, the mount options should be the same on the metadata server and all potential metadata servers. For example, you can create a `samfs.cmd` file that contains the mount options and copy that file to all of the hosts.

For more information about mounting shared file systems, see Mount Options in a Shared File System and the `mount_samfs`(1M) man page. For more information about mounting and unmounting file systems, see Performing Operations.

How to Mount a Shared File System

1. Become superuser on the metadata server and on all the client hosts.
2. Use the `mount`(1M) command to mount the metadata server.
   Mount the file system on the metadata server before you mount the file system on any client hosts. For example:

   ```
   # mount -F samfs qfs1 /qfs -o shared
   ```

   > Tip -
   > If the mount information has been placed in `/etc/vfstab`, you can use the simpler command:
   > `# mount qfs1`
   > For information about the `/etc/vfstab` file and other mount options, see Setting Up Mount Parameters.

3. Use the `mount`(1M) command to mount the client hosts.
   You can mount the file system on the client hosts in any order. For more information about the `mount`(1M) command, see the `mount`(1M) man page.

How to Unmount a Shared File System

> Note -
> If the file system is shared through NFS or SAMBA, unshare the file system before you unmount it.

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

   ```
   client# umount /samqfs
   ```

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

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

2. Unmount the file system on the metadata server:

   ```
   metaserver# umount /samqfs
   ```

   At unmounting time, several conditions can be present in a file system that may require you to issue the `umount`(1M) command a second time.

   > Note -
   > If the file system still does not unmount, use `unshare`(1M), `fuser`(1M), or another command in conjunction with the `umount`(1M) command.

For more information on unmounting procedures, see the umount(1M) man page and Unmounting a File System.
You can also use the -o await_clients # flag with the umount command. This causes the unmount process to wait a specified number of seconds (#) for clients to unmount. At the end of the waiting period, or as soon as all clients have unmounted, the unmount proceeds. If this argument is specified for a non-shared file system, or if the host is not the metadata server for the shared file system, the option will be ignored.
This flag can also be used in conjunction with the -f option. In this case, the software will wait for the specified time period before forcing the unmount.

Adding or Removing a Client Host

The following subsections provide instructions for adding and removing client host systems in a shared file system.

How to Add a Client Host to a Shared File System

You can add a client host to a shared file system after you have configured and mounted the file system on all participants.

1. Become superuser on the metadata server.
2. Use the samsharefs(1M) command to retrieve the current shared file system information and write it to an editable file.
   - If the shared file system is mounted, issue the samsharefs(1M) command on the current metadata server. For example:

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

   - If the shared file system is not mounted, issue the samsharefs(1M) command with its -R option from the metadata server or from any of the potential metadata servers. For example:

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

     You can only issue the samsharefs(1M) command on the active metadata server or on client hosts configured as potential metadata servers. For more information, see the samsharefs(1M) man page.
3. Use vi(1) or another editor to open the shared file system information file.
   The following code example shows this step.

   ```
   # vi /etc/opt/SUNWsamfs/hosts.sharefs1
   # File /etc/opt/SUNWsamfs/hosts.sharefs1
   # Host     Host IP     Server     Not     Server
   # Name     Addresses     Priority     Used     Host
   # ----     --------------------------------     --------     ----     -----
   titan    172.16.0.129     1     -     server
   tethys    172.16.0.130     2    -
   mimas    mimas     -    -
   dione    dione     -    -
   ```

4. Use the editor to add a line for the new client host.
   The following code example shows the file after addition of the line for helene as the last line.

   ```
   # File /etc/opt/SUNWsamfs/hosts.sharefs1
   # Host     Host IP            Server      Not      Server
   # Name     Addresses          Priority    Used     Host
   # ----     ---------------    --------    ----     -----
   titan      172.16.0.129         1           -      server
   tethys     172.16.0.130         2           -
   mimas      mimas                -           -
   dione      dione                -           -
   helene     helene               -           -
   ```

5. Use the samsharefs(1M) command to update the current information in the binary file.
   The options to use with this command, and the system from which this command is issued, depend on whether the Sun QFS shared file system is mounted, as follows:

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

```
# samsharefs -u sharefs1
```

- If the file system is not mounted, issue the `samsharefs`(1M) `-R -u` command from the active metadata server or from any of the potential metadata servers. For example:

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

  The client host `helene` is now recognized.

6. As superuser, log in to the client host to be added.
7. Use the `format`(1M) command to verify the presence of client host disks.
8. Update the `mcf` file on the client host.
   Before a host system can access or mount a shared file system, it must have that file system defined in its `mcf` file. The `mcf` file must be updated to match all client hosts in the shared file system. The file system and disk declaration information must have the same data for the Family Set Name, Equipment Number, and Equipment Type as the configuration on the metadata server. The `mcf` files on the client hosts must also include the shared keyword. The device names, however, can change, since controller assignments will probably differ from host to host.
   For information on how to edit the `mcf` file, see Updating the mcf file in a Sun QFS Shared Environment.
9. Issue the `samd`(1M) `config` command on the metadata server host:

```
# samd config
```

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

10. (Optional) Create the local hosts configuration file on the new client host.
    You might want to perform this step if your Sun QFS shared host systems have multiple host interfaces. The local hosts configuration file defines the host interfaces that the metadata server and the client hosts can use when accessing the file system. You use this file to specify how file system traffic should flow over public and private networks in your environment.
    For information on creating the local hosts file, see Creating the Local Hosts Configuration File.
11. Issue the `samd`(1M) `config` command on the client host:

```
# samd config
```

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

12. Verify that the `sam-sharefsd` daemon is running for this file system.
    To accomplish this, use the `ps`(1) and `grep`(1) commands as shown in the following code example.

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

    The code example above shows that the `sam-sharefsd` daemon is active for the `sharefs1` file system. If the output returned on your system does not show that the `sam-sharefsd` daemon is active for your Sun QFS shared file system, perform the diagnostic procedures described in the SAM-QFS Troubleshooting Guide.

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

```
# mkdir /sharefs1
```

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

For example:

```
# chmod 755 /sharefs1
```

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

15. Modify the `/etc/vfstab` file.
You must have an entry in the `/etc/vfstab` file for the Sun QFS shared file system. Specify `shared` in the Mount Parameters field. In addition, do one of the following:
    - If you do not want to mount this file system automatically at boot time, type no in the `Mt@boot` field.
    - If you do want the Sun QFS shared file system to automatically mount at boot, do the following:
        - Type yes in the `Mt@boot` field.
        - Add the bg mount option in the `Mt params` field. The bg mount option mounts the file system in the background if the metadata server is not responding.
The following code example shows the `shared` and bg entries in the `Mt params` field.

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

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

```
# df -k
```

The file system should be included in the displayed list.

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

```
# mount /sharefs1
```

For more information about mounting Sun QFS shared file systems, see Mount Options in a Shared File System, or see the `mount_samfs`(1M) man page.

How to Remove a Client Host From a Shared File System

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

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

2. Use the `umount`(1M) command to unmount the shared file system on each client host on which the shared file system is mounted.
For example:

```
client# umount sharefs1
```

3. Use the `umount`(1M) command to unmount the shared file system on the metadata server.
For example:

```
metaserver# umount sharefs1
```

4. If you have not already done so, log in as superuser to the metadata server for the shared file system.
5. Use the samsharefs(1M) command to obtain the current configuration information.
   The following example command writes current configuration information to file /etc/opt/SUNWsamfs/hosts.sharefs1:

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

6. Use vi(1) or another editor to open the shared file system information file.
   The following code example shows the file before the client host is deleted.

```
# vi /etc/opt/SUNWsamfs/hosts.sharefs1
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host    Host IP         Server      Not     Server
# Name    Addresses       Priority    Used    Host
# ----    ------------    --------    ----    -----
titan     172.16.0.129      1          -      server
tethys    172.16.0.130      2          -
mimas     mimas             -          -
dione     dione             -          -
helene    helene            -          -
```

7. Use the editor to delete the client host or hosts that are no longer to be supported.
   The following code example shows the file after the line for helene has been deleted.

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host    Host IP          Server      Not     Server
# Name    Addresses        Priority    Used    Host
# ----    --------------   --------    ----    -----
titan     172.16.0.129        1          -    server
tethys    172.16.0.130        2          -
mimas     mimas               -          -
dione     dione               -          -
```

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

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

   The host helene has been removed.

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

```
# samsharefs -R sharefs1
```

10. Use the mount(1M) command to mount the shared file system, first on the metadata server and then on each client host in the file system.
    For information about the mount(1M) command, see the mount_samfs(1M) man page.

Updating the mcf file in a Shared File System Environment

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

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

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

Example – ***samfsconfig***(1M) Command Example on ***tethys***

The following example shows how the `samfsconfig`(1M) command is used to retrieve device information for family set `sharefs1` on client `tethys`. Because `tethys` is a potential metadata server, it is connected to the same metadata disks as `titan`, another metadata server in the shared file system.

```
tethys# samfsconfig /dev/dsk/*
#
# Family Set &rdquo;sharefs1&rsquo; Created Wed Jun 27 19:33:50 2003
#
sharefs1                         10 ma sharefs1 on shared
/dev/dsk/c2t50020F23000065EEd0s6 11 mm sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12 mr sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13 mr sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14 mr sharefs1 on
```

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

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

The next example shows the resulting `mcf` file.

Example – ***mcf*** File for ***sharefs1*** Client Host ***tethys***

```
# Equipment                      Eq  Eq   Family   Dev   Add
# Identifier                     Ord Type Set      State Params
# ----------                     --- ---- ------   ----- ------
sharefs1                         10  ma   sharefs1 on    shared
/dev/dsk/c2t50020F23000065EEd0s6 11  mm   sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12  mr   sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13  mr   sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14  mr   sharefs1 on
```

Example – ***samfsconfig***(1M) Command Example on ***mimas***

The following example shows how the `samfsconfig`(1M) command is used to retrieve device information for family set `sharefs1` on client host `mimas`. In this example, `mimas` can never become a metadata server, and it is not connected to the metadata disks.

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

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

- Create a file system Family Set declaration line, beginning with `sharefs1`, in the `mcf` file for client host `mimas`. Enter the `shared` keyword in the Additional Parameters field of the file system Family Set declaration line.

- Create one or more nodev lines for each missing Equipment Number entry. For each of these lines, the keyword nodev must appear in the Equipment Identifier field for the inaccessible device.
- Ensure that each Device State field is set to on.
- Uncomment the device lines.

The next example shows the resulting mcf file for mimas.

Example – ***mcf*** File for Client Host ***mimas***

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

Creating the Local Hosts Configuration File

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

```
/etc/opt/SUNWsamfs/hosts._family-set-name_.local
```

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

The following table shows the fields in the local hosts configuration file.

Table – Local Hosts Configuration File Fields

Field	Content
Host Name	This field must contain the alphanumeric name of a metadata server or potential metadata server that is part of the Sun QFS shared file system.
Host Interfaces	This field must contain a comma-separated list of host interface addresses. This field can be created from the output received from the ifconfig(1M) -a command. The individual interfaces can be specified in one of the following ways:* Dotted-decimal IP address form* IP version 6 hexadecimal address form* As a symbolic name that the local domain name service (DNS) can resolve to a particular host interface Each host uses this field to determine whether it will try to connect to the specified host interface. The system evaluates the addresses from left to right, and the connection is made using the first responding address in the list that is also included in the shared hosts file.

In a shared file system, each client host obtains the list of metadata server IP addresses from the metadata server host.

The metadata server and the client hosts use both the /etc/opt/SUNWsamfs/hosts.fsname file on the metadata server and the hosts.fsname.local file on each client host (if it exists) to determine the host interface to use when accessing the file system. This process is as follows:

> Note -
> Client, as in network client, is used to refer to both client hosts and the metadata server host.

The client obtains the list of metadata server host IP interfaces from the file system's on-disk host file.
To examine this file, issue the samsharefs(1M) command from the metadata server or from a potential metadata server.
The client searches its files for a hosts.fsname.local file.
Depending on the outcome of the search, one of the following courses of action is taken:

-
    - If a `hosts.`fsname`.local` file does not exist, the client attempts to connect, in turn, to each address in the system hosts configuration file until it succeeds.
    - If the `hosts.`fsname`.local` file exists, the client performs the following:
      It compares the list of addresses for the metadata server from both the `/etc/opt/SUNWsamfs/hosts.`fsname file on the metadata server and the `hosts.`fsname`.local` file.
      It builds a list of addresses that are present in both places, and then it attempts to connect to each of these addresses, in turn, until it succeeds. If the order of the addresses differs in these files, the client uses the ordering in the `hosts.`fsname`.local` file.

Example – Sun QFS Shared File System Hosts File Example

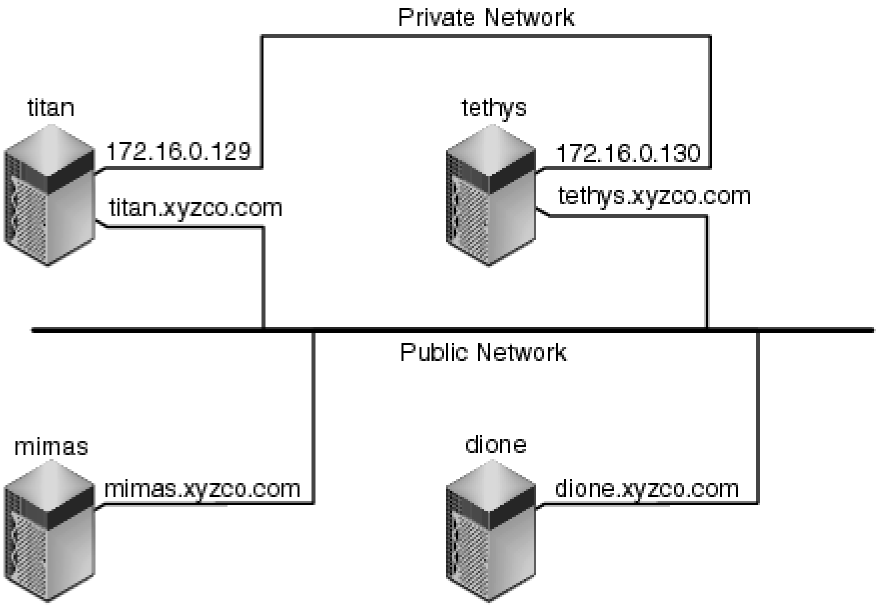
This set of examples shows a detailed scenario for a shared file system that comprises four hosts.

The following example shows a hosts file that lists four hosts.

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host  Host IP           Server     Not    Server
# Name  Addresses         Priority   Used   Host
# ----  ----------------- --------   ----   -----
titan   172.16.0.129        1          -    server
tethys  172.16.0.130        2          -
mimas   mimas               -          -
dione   dione               -          -
```

The following figure shows the interfaces to these systems.

Figure – Network Interfaces for Sun QFS Shared File System Hosts File Example



Example – File ***hosts.sharefs1.local*** on ***titan*** and ***tethys***

Systems `titan` and `tethys` share a private network connection with interfaces `172.16.0.129` and `172.16.0.130`. To guarantee that `titan` and `tethys` always communicate over their private network connection, the system administrator has created identical copies of `/etc/opt/SUNWsamfs/hosts.sharefs1.local` on each system.

The following example shows the information in the `hosts.sharefs1.local` files on `titan` and `tethys`.

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

Example – File ***hosts.sharefs1.local*** on ***mimas*** and ***dione***

Systems mimas and dione are not on the private network. To guarantee that they always connect to titan and tethys through titan's and tethys's public interfaces, the system administrator has created identical copies of /etc/opt/SUNWsamfs/hosts.sharefs1.local on mimas and dione.

The following example shows the information in the hosts.sharefs1.local files on mimas and dione.

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

Changing the Metadata Server in a Shared File System Environment

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

You can change the metadata server system manually under the following circumstances:

- If the metadata server becomes unavailable
- If you want to change the metadata server or the potential metadata servers

For a metadata server change to succeed, the mount options of the existing metadata server and all potential metadata servers must be the same.

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

- How to Change the Metadata Server When the Metadata Server Is Available
- How to Change the Metadata Server When the Metadata Server Is Unavailable

How to Change the Metadata Server When the Metadata Server Is Available

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

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

> Note -
> In archiving environments, you should stop all archiving operations on the metadata server before you issue this command.

How to Change the Metadata Server When the Metadata Server Is Not Available

If the metadata server of a shared file system crashes, it is safe to change the metadata server only after rebooting the metadata server or otherwise ensuring that the server cannot issue any I/O before being rebooted. Do not use any of the following methods to stop the server, because these are likely to corrupt the file system:

- Issuing an L1-A key sequence
- Performing an involuntary failover to another host
- Issuing a go (continue) command, requesting a dump file, or issuing a sync command to the old metadata server
  Similarly, if the metadata server panics and drops into kernel adb(1), do not change the metadata server and then issue a :c (continue) command on the server. This action causes the old metadata server to push stale buffers out to the now-active file system.

Use the following steps to change the metadata server:

Steps

1. Ensure that the existing metadata server cannot restart without being rebooted.
Specifically, ensure that the server is powered down, rebooted, halted, or disconnected from the metadata disks. Your goal is to bring down the old metadata server and flush or destroy all buffers (or otherwise ensure that they cannot be rewritten).
The following code example shows the key sequence to use from the kadb prompt.

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

The following code example shows the key sequence to use from the PROM prompt.

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

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

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

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

The wait ensures that all client leases expire before you issue the samsharefs(1M) command. If you are uncertain as to whether the lease time has expired, bring up the samu(1M) N display. For information about samu(1M), see Appendix B, Using the samu(1M) Operator Utility. For information about leases and their durations, see Using Leases in a Sun QFS Shared File System: the rdlease=n, wrlease=n, and aplease=n Options.

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

3. (Optional) Unmount the file system.
Perform this step only if you want to perform a file system check.
Use the procedure in Unmounting a File System.
4. (Optional) Issue the samfsck(1M) command to perform a file system check.
If the metadata server of a Sun QFS shared file system crashes, the server should be rebooted and the file system should be unmounted on all clients before samfsck(1M) is run. The server and clients preallocate blocks before changing the length of files. The samfsck (1M) command cleans up files that have extra blocks allocated, and these extra blocks might contain data. If such a cleaned-up file is awaiting a size update from the client, the file will be missing those blocks when the client continues. As a result, the file will be missing data, and the missed data will read as zeroes.

Changing the Metadata Server in an Archiving Environment

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

You can change the metadata server system manually under the following circumstances:

- If the metadata server becomes unavailable
- If you want to change the metadata server or the potential metadata servers

For a metadata server change to succeed, the mount options of the existing metadata server and all potential metadata servers must be the same.

How to Change the Metadata Server in an Archiving Environment

Archiving functions can only run on one host at any time. This procedure assumes that both systems are up at the time of the transfer. In this example, we move archiving functions from host A to host B.

Before carrying out this procedure, verify that host B has access to the robot catalog from host A. The `archiver.cmd` file, `mcf` file, `stager.cmd` file, and other configuration files must be identical to those on host A.

Steps

1. Idle archiving processes on host A by carrying out the following steps:
   a. Run `samcmd aridle` and `samcmd stidle` to halt archiving and staging on host A.
   These commands will allow current archiving and staging to complete, but will not start any new work.
   b. Idle all of the tape drives on host A.
   This can be done with `samcmd` eq `idle`, where eq is the equipment number of the drive. This will put the drives in an “off” state after any current I/O completes.
   c. When the archiver and stager are idle and the tape drives are all in the “off” state, run the `samd stop` command to halt all of the robot and tape-related daemons.
   d. If you have a `cron` job that runs the recycler, remove this entry from the crontab and verify that the recycler is not currently running.
   At this point, the archiving processes have been halted and file system failover to host B can be performed.
2. Start the archiving processes on host B by running `samd config` on host B.
   This causes `sam-fsd` and its subprocesses (archiver, stager, and so on) to reconfigure and re-read the configuration files. It also causes `sam-amld` and the tape-library-related daemons to start. At this point all Sun QFS shared client applications waiting for stages must reissue the stage requests.
   Host B should now be fully functioning as the archiving process server and metadata server for all file systems.

Converting an Unshared File System to a Shared File System

To perform initial installation and configuration for a shared file system, follow the instructions in Installing Sun QFS. Many examples in this section use host names and configuration information that were introduced in that document.

To convert an unshared file system to a shared file system consists of two tasks:

- First, convert the metadata server.
- Second, add each client to the metadata server. This section describes these procedures.

How to Convert an Unshared Metadata Server to a Shared Metadata Server

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

1. As superuser, log in to the system to be used as the primary metadata server.
2. (Optional) Back up all site-customized system files and configuration files.
   Depending on your software, these files might include `mcf`, `archiver.cmd`, `defaults.conf`, `samfs.cmd`, or `inquiry.conf`. Back up these files for all file systems. Also make sure that you have backup copies of files in the `/etc/opt/SUNWsamfs` directory, and files in the `/var/opt/SUNWsamfs` directory.
3. Ensure that each file system to be modified is backed up.
   File systems should be backed up regularly according to your site's policies. If you are comfortable with the backup files that already exist for your file systems, there is no need to back them up again now.
4. Use the `umount`(1M) command to unmount the file system.
   For instructions, see Unmounting a File System.
5. Use the `samfsck`(1M) `-S -F` family-set-name command to convert the file system to a Sun QFS shared file system.
   For family-set-name, specify the family set name of the file system that you are converting to a new shared file system. For example:

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

6. Edit the `/etc/opt/SUNWsamfs/mcf` file to add the `shared` keyword in the file system’s Additional Parameters field.
   For example:

```
# Equipment Eq Eq Family Dev Add
# Identifier Ord Type Set State Params
# ---------- --- ---- ------ ----- ------
sharefs1 10 ma sharefs1 on shared
/dev/dsk/c2t50020F23000065EEd0s6 11 mm sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12 mr sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13 mr sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14 mr sharefs1 on
```

7. Edit the `/etc/vfstab` file to add the `shared` keyword in the file system's Mount Parameters field.
   For example:

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

8. Create the `/etc/opt/SUNWsamfs/hosts.`fsname hosts configuration file.
   For example:

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host     Host      IP                Server    Not  Server
# Name       Addresses                 Priority  Used Host
# ---- ------------------------------ -------- ---- -----
titan    titan-ge                       0 1 - server
tethys   tethys-ge                      0 2 - server
```

   See Creating the Shared Hosts File for more information about creating the hosts configuration file.

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

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

> Note -
> If you see an error message from this command, you can probably ignore the message.

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

```
# samd config
```

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

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

How to Add a Client to the Metadata Server

1. Use the `mkdir`(1) command to create the mount point for the file system.
   For example:

```
# mkdir /sharefs1
```

2. (Optional) Create an `/etc/opt/SUNWsamfs/hosts.`file-system-name`.local` local hosts configuration file.

You might want to perform this step if your Sun QFS shared host systems have multiple host interfaces. The local hosts configuration file defines the host interfaces that the metadata server and the client hosts can use when accessing the file system. You use this file to specify how file system traffic should flow over public and private networks in your environment.
The following code example shows a sample local hosts configuration file.

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

For more information on creating the local hosts file, see Creating the Local Hosts Configuration File.

3. If you want to move files from an existing Sun QFS file system into a new Sun QFS shared file system, ensure that each file system to be modified is backed up.
   File systems should be backed up regularly according to your site's policies. If you are comfortable with the backup files that already exist for your file systems, there is no need to back them up again now.
4. Use the `umount(1M)` command to unmount the file system.
   For instructions, see Unmounting a File System.
5. Edit the `/etc/vfstab` file to add the `shared` keyword in the file system's Mount Parameters field.
   For example:

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

6. Create the `/etc/opt/SUNWsamfs/hosts.`fsname hosts configuration file.
   The following code example shows a sample.

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IP Server Not Server
# Name Addresses Priority Used Host
# ---- --------- -------- ---- -----
titan titan-ge0 1 - server
tethys tethys-ge0 2 - server
```

For more information about creating the hosts configuration file, see Creating the Shared Hosts File.

Converting a Shared File System to an Unshared File System

To convert a Sun QFS shared file system to an unshared Sun QFS file system requires two tasks:

- Remove the shared clients.
- Convert the metadata server.

This section describes these procedures.

How to Remove a Client From a Shared File System

1. Use the `umount(1M)` command to unmount the file system.
   For instructions, see Unmounting a File System.
2. Delete the file system's entry from the `/etc/opt/SUNWsamfs/mcf` file.
3. Delete the file system's entry from the `/etc/vfstab` file.
4. Run the `samd`(1M) `config` command:

```
# samd config
```

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

5. Delete the mount point for the file system.

How to Convert a Shared Metadata Server to an Unshared System

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

1. As superuser, log in to the metadata server system.
2. Back up all site-customized system files and configuration files.
   Depending on your software, these files might include mcf(4), archiver.cmd, defaults.conf, samfs.cmd, inquiry.conf, and so on. Back up these files for all file systems. Also make sure that you have backup copies of files in the /etc/opt/SUNWsamfs directory and files in the /var/opt/SUNWsamfs directory.
3. If you want to move files from an existing Sun QFS shared file system into a new Sun QFS file system, ensure that each file system to be modified is backed up.
   File systems should be backed up regularly according to your site's policies. This is described as the last step in the installation procedure. If you are comfortable with the backup files that already exist for your file systems, there is no need to back them up again now.
4. Use the umount(1M) command to unmount the file system.
   For instructions, see Unmounting a File System.
5. Run the samfsck(1M) -F -U file-system-name to convert the Sun QFS shared file system to an unshared file system.
   For file-system-name, specify the name of the Sun QFS shared file system that you are converting to a new unshared file system. For example:

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

6. Edit the /etc/opt/SUNWsamfs/mcf file to remove the shared keyword from the file system's Additional Parameters field.
   For example:

```
# Equipment Eq Eq Family Dev Add
# Identifier Ord Type Set State Params
# ---------- --- ---- ------ ----- ------
samfs1 10 ma samfs1 on
/dev/dsk/c2t50020F23000065EEd0s6 11 mm samfs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12 mr samfs1 on
/dev/dsk/c7t50020F2300006099d0s6 13 mr samfs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14 mr samfs1 on
```

7. Edit the /etc/vfstab file to remove the shared keyword from the file system's Mount Parameters field.
   For example:

```
# File /etc/vfstab
# FS name FS to fsck Mnt pt FS type fsck pass Mt@boot Mt params
samfs1 - /samfs1 samfs - no
```

8. Delete the /etc/opt/SUNWsamfs/hosts.file-system-name configuration file.
9. Run the samd(1M) config command:

```
# samd config
```

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

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

Client-Server Communications in a Shared File System

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

The system generates the following messages to describe status conditions:

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

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

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

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

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

When the shared file system mounts on the server, it generates the following message:

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

Because the metadata server looks up file names on behalf of all clients, performance can be slow with the default size of the Solaris directory name lookup cache (DNLC) on the metadata server. To increase performance when clients are frequently opening a large number of files, you might want to double or even triple the size of this cache from its default.

This procedure is documented in the Solaris Tunable Parameters Reference Manual. The parameter that controls the size of the directory name lookup cache is `ncsize`.

For more information about troubleshooting, see SAM-QFS Troubleshooting.

Adding Disk Cache to a File System

To increase the disk cache for a file system, you add disk partitions or disk drives, and then update the `mcf` file and use the `samgrowfs`(1M) command to expand the file system. You do not need to reinitialize or restore the file system.

When making changes to the `mcf` file, be aware of the following:

- You can configure up to 252 disk partitions in a file system.
- To increase the size of a Sun QFS file system, you must add at least one new metadata partition. Metadata partitions require an Equipment Type value of `mm`.
- If you want to add new partitions for metadata or for data, add them to the `mcf` file after the existing disk partitions.
- Do not change the Equipment Identifier name in the `mcf` file. If the name in the `mcf` file does not match the name in the superblock, the file system can no longer be mounted. Instead, the following message is logged in `/var/adm/messages`:

```
WARNING SAM-FS superblock equipment identifier <id> on eq <eq>
does not match <id> in mcf
```

How to Add Disk Cache to a File System

1. Use the `umount`(1M) command to unmount the file system you want to expand.
   If the file system is shared, unmount the file system on all client hosts and then on the metadata server. You can then perform the remaining steps in this procedure on the metadata server.
   For more information about unmounting a file system, see Unmounting a File System.

2. If you want to rename the file system during this procedure, use the `samfsck`(1M) command with its `-R` and `-F` options to rename the file system.
   For more information about this command, see the `samfsck`(1M) man page.
3. Edit the `/etc/opt/SUNWsamfs/mcf` file to add the disk cache.
4. Issue the `sam-fsd`(1M) command to check for errors in the `mcf` file:

```
# sam-fsd
```

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

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

```
# samd config
```

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

6. Issue the `samgrowfs`(1M) command on the file system that is being expanded.
   For example, type the following command to expand file system `samfs1`:

```
# samgrowfs samfs1
```

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

7. Mount the file system.
   For information about mounting a Sun QFS file system, see the `mount_samfs`(1M) man page.
8. If the file system is a Sun QFS shared file system, edit the `mcf` file on each participating client host to match the metadata server's `mcf` file.

Recreating a File System

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

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

This section describes this procedure.

How to Back Up and Re-create a File System

1. Back up all site-customized system files and configuration files.
   Depending on your software, these files might include `mcf`, `archiver.cmd`, `defaults.conf`, `samfs.cmd`, or `inquiry.conf`. Back up these files for all file systems in your Sun QFS environment. Also make sure that you have backup copies of files in the `/etc/opt/SUNWsamfs` directory, files in the `/var/opt/SUNWsamfs` directory, and shared hosts files.
2. Ensure that each file system to be modified is backed up.
   File systems should be backed up regularly according to your site's policies. If you are comfortable with the backup files that already exist for your file systems, there is no need to back them up again now. If, however, you need to back up your file systems to preserve information created since the last dump file was created, do so now. For information about how to create a dump file using `qfsdump`, see Backing Up SAM-QFS Data and Files.
3. Unmount the file system.
   For instructions, see Unmounting a File System.
4. If you want to rename the file system during this procedure, use the `samfsck`(1M) command with its `-R` and `-F` options.
   For more information, see the `samfsck`(1M) man page.
5. Edit the `/etc/opt/SUNWsamfs/mcf` file to add, change, or remove partitions.
   For more information, see Adding Disk Cache to a File System.
6. Type the `sam-fsd`(1M) command to check for errors in the `mcf` file:

```
# sam-fsd
```

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

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

```
# samd config
```

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

8. Issue the sammkfs(1M) command to re-create the file system.
For example, the following command creates samfs10:

```
# sammkfs samfs10
```

9. Issue the mount(1M) command to mount the file system.
For information about mounting a Sun QFS file system, see the mount_samfs(1M) man page.
10. Issue the cd(1) command to change to the mount point of the file system.
11. Use the qfsrestore(1M) command, or use SAM-QFS Manager, to restore each file.
Restore from the dump file you had or from the dump file created in Step 1.
For more information, see the qfsdump(1M) man page or the SAM-QFS Manager online help.

Administering File System Quotas

About File System Quotas

File system quotas control the amounts of online and total disk space that can be consumed by a specific user, by a group of users, or by a site-determined group of users called an admin set.

Quotas help control the size of a file system by limiting the amount of space and the number of inodes that each user can consume. Quotas can be especially useful on file systems that contain user home directories. After quotas are enabled, you can monitor usage and adjust the quotas as needs change.

A file system provides a user with blocks for data and inodes for files. Each file uses one inode, and file data is stored in a disk allocation unit (DAU). DAU sizes are determined at the time the file system is created. Quotas account for disk usage in multiples of 512 bytes.

The following subsections provide background information about using quotas.

Types of Quotas, Quota Files, and Quota Records

You can set quotas according to user ID, group ID, or an administrator's site-specific grouping. This site-specific grouping is called an admin set ID. You can use an admin set ID, for example, to identify a collection of users working on a project for which file system quotas are imposed.

Quotas are enabled when the quota mount option is in effect and the system detects the presence of one or more quota files in the file system's root directory. The quota mount option is enabled by default. If you mount the file system with noquota in effect, quotas are disabled. For more information about mount options, see the mount_samfs(1M) man page.

Each quota file contains a sequence of records. Record zero is the record for the system administrator's quotas and resource usage. System administrator quotas are never enforced, but you can use any record, including the system administrator's record, as a template for subsequent records in the quota file. For more information about this practice, see How to Enable or Change Limits for Users, Groups, or Admin Sets Using an Existing Quota File.

Record one is the record in the quota file for user one, group one, or admin set ID one, depending on the type of quota file. You can edit record one and all subsequent records in order to set different quotas for different users. The following table shows the quota file names and the quotas they enable in /root.

Table – Quota File Names

Quota File Name	Quota Type
`.quota_u`	UID (system user ID)
`.quota_g`	GID (system group ID)
`.quota_a`	AID (system admin set ID)

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

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

(highest-ID + 1) x 128 = xx / 4096 = zero quota file size

Soft Limits and Hard Limits

You can set both soft and hard limits. A hard limit specifies a fixed amount of system resources available for use, which the system never allows a user to exceed. A soft limit specifies a level of system resource use that can be exceeded temporarily, up to the hard limit. The soft limit is never larger than the hard limit.

If a user attempts to allocate resources beyond the hard limit, the operation is aborted. In this case, the operation fails and generates an EDQUOT error.

After a user exceeds a soft limit, a timer starts, and the user enters a grace period. While the timer is ticking, the user is allowed to operate above the soft limit. After the user goes below the soft limit, the timer is reset. If the grace period ends and the timer stops without the user's having gone below the soft limit, the soft limit is then enforced as a hard limit.

For example, assume that a user has a soft limit of 10,000 blocks and a hard limit of 12,000 blocks. If the user's block usage exceeds 10,000 blocks and the timer exceeds the grace period, this user is no longer able to allocate more disk blocks on that file system until usage drops below the 10,000-block soft limit.

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

Disk Blocks and File Limits

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

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

Enabling Quotas

You can enable quotas through a process that includes editing system files, creating quota files, and entering various quota commands.

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

Guidelines for Setting Up Quotas

Before you enable quotas, you should determine how much disk space and how many inodes to allocate to each user. If you want to be sure that the total file system space is never exceeded, you can divide the total size of the file system by the number of users. For example, if three users share a 100-megabyte slice and have equal disk space needs, you could allocate 33 megabytes to each. In environments in which not all users are likely to reach their limits, you might want to set individual quotas so that they add up to more than the total size of the file system. For example, if three users shared a 100-megabyte slice, you could allocate 40 megabytes to each.

You can use the following quota commands, in the formats shown, for displaying quota information:

- The squota(1) command is for end users. It enables them to retrieve quota information for themselves on a user, group, or admin set basis.
- The samquota(1M) command is for system administrators. It enables you to retrieve quota information or to set quotas. Use the -U, -G, and -A options for a user, a group, or an admin set, respectively. The following example shows this.

Example – Using ***samquota***(1M) to Retrieve Information

The first command displays a user quota. The second command shows a group quota. The last command shows the quota for an admin set.

```
# samquota -U janet </mount-point>
# samquota -G pubs </mount-point>
# samquota -A 99 </mount-point>
```

How to Configure a New File System to Use Quotas

Use this procedure if you are creating a new file system and no files currently reside in the file system. To configure an existing file system to use quotas, see To Configure an Existing File System to Use Quotas.

Before you start this procedure, make sure that you do not have the noquota mount option specified in your samfs.cmd or /etc/vfstab files.

1. Become superuser.
2. Create the file system.
   Either follow the steps outlined in the Sun SAM-QFS Installation and Upgrade Guide, or use the examples in Configuration Examples to create the mcf file, create the mount point, initialize the file system, and so on .
3. Use the mount(1M) command to mount the file system.
   For example:

   ```
   # mount /qfs1
   ```

4. Use the dd(1M) command to create the quota files.
   The arguments to this command depend on the type of quota you are creating, as follows:
   - To create admin set quotas, use the following command:

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

   - To create group quotas, use the following command:

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

   - To create user quotas, use the following command:

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

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

5. Use the umount(1M) command to unmount the file system in which the quota files have been created. For example:

   ```
   # umount /qfs1
   ```

   The file system must be unmounted so it can be remounted and have its quota files read at mount time. For more information, see the umount(1M) man page.

6. Use the samfsck(1M) command to perform a file system check.
   In the following example, the -F option resets the in-use values in the quota files:

```
# samfsck -F qfs1
```

7. Use the mount(1M) command to remount the file system.
   The system enables quotas when it detects the presence of one or more quota files in the root directory.

   > Note -
   > You do not need to include the quota mount option in the /etc/vfstab or samfs.cmd file. The quota mount option is enabled by default with the mount(1M) command, and quotas are enabled automatically when the system detects the presence of quota files.

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

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

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

How to Configure an Existing File System to Use Quotas

Use this procedure if you are creating quotas for a file system that is already populated with files. If you are configuring a new file system to use quotas, see To Configure a New File System to Use Quotas.

Before you start this procedure, make sure that you do not have the noquota mount option specified in your samfs.cmd or /etc/vfstab files.

1. Use the su(1) command to become superuser.
2. Use the mount(1M) command to examine the /etc/mnttab file and ensure that the file system is mounted:

```
# mount
```

   Make sure that the file system is listed in the mount list that is displayed.

3. Use the cd(1) command to change to the root directory of the file system for which quotas are to be enabled. For example:

```
# cd /oldfs1
```

4. Use the ls(1) -a command to retrieve the list of files in this directory and verify that quotas do not already exist on the file system.
   If any of the following files are present, quotas have been enabled for this file system: .quota_u, .quota_g, .quota_a.
   If any quota type is established for a file system, you can establish any other quota type later. Be careful not to modify existing quota files when adding new ones.
5. If the quota files do not exist for the types of quotas you wish to enforce, use the dd(1M) command to create the quota files.
   Determine the highest existing ID numbers of the types of quotas you wish to enforce. Make the initial, zero, quota files large enough to hold the records for those IDs; each quota file record requires 128 bytes.
   For example, if you want to enable admin set quotas, and the highest admin set ID in use on the file system is 1024, the calculation is as follows:
   (1024 + 1) x 128 = 131200
   131200/4096 = 32.031...
   Use the following command:

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

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

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

```
# umount /oldfs1
```

The file system must be unmounted so it can be remounted and have its quota files read at mount time. For more information about unmounting a file system, see Unmounting a File System.

7. Use the `samfsck`(1M) `-F` command to perform a file system check. This command updates records allocated in the quota files with current usage information. For example:

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

8. Use the `mount`(1M) command to remount the file system in which the quota files have been created.
   The system enables quotas when it detects the presence of one or more quota files in the `/root` directory.
   You do not need to include the `quota` mount option in the `/etc/vfstab` or `samfs.cmd` file. The `quota` mount option is enabled by default with the `mount`(1M) command, and quotas are enabled automatically when the system detects the presence of quota files.

> Note -
> If quota files are present and if the file system is mounted with quotas disabled, the quota records become inconsistent with actual usages when blocks or files are allocated or freed. If a file system with quotas is mounted and run with quotas disabled, run the `samfsck`(1M) `-F` command to update the quota file usage counts before again remounting the file system with quotas enabled.

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

9. Use the `samquota`(1M) command to set quotas for users, groups, or admin sets.
   Subsequent sections in this chapter provide procedures and show examples of this process. For more information about the `samquota`(1M) command, see the `samquota`(1M) man page.

How to Assign Admin Set IDs to Directories and Files

1. Use the `su`(1) command to become superuser.
2. Set the admin IDs. Use the `samchaid`(1M) command to change the admin set IDs for the directory or file, as follows:
   - To set IDs for a file or directory, specify the directory name or path. For example:

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

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

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

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

Setting Infinite Quotas

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

How to Set an Infinite Quota

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

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

You can set infinite quotas for particular users, groups, or admin set IDs by setting zero values for all hard and soft limits. The following example shows how to set infinite quotas.

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

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

Enabling Default Quota Values

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

How to Enable Default Quota Values for Users, Groups, or Admin Sets

- Use the `samquota`(1M) command. The following command sets default quotas for all admin set IDs:

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

The preceding command sets any user's uninitialized admin set quota limits as follows:

- The soft online block limit (`-b` limit`:s`) is set to 12,000 blocks.
- The hard online block limit (`-b` limit`:h`) is set to 15,000 blocks.
- The total soft block limit (`-b` limit`:s:t`) is set to 12 gigablocks.
- The total hard block limit (`-b` limit`:h:t`) is set to 15 gigablocks.
- The soft file limit (`-f` limit`:s`) is set to 1000 files.
- The hard file limit (`-f` limit`:h`) is set to 1200 files.
- The grace period (`-t` limit) is set to one week.

> Note -
> If a quota record already exists, the existing values remain in effect. This occurs, for example, if the admin group already has blocks assigned to it.

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

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

Enabling Limits

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

How to Enable Limits for Users, Groups, or Admin Sets

- Use the `samquota`(1M) command. For example, the following commands enable limits for users, groups, and admin sets, respectively.

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

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

How to Enable or Change Limits for Users, Groups, or Admin Sets Using an Existing Quota File

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

1. Retrieve a quota file and direct the output to a temporary file.
   Use the `samquota`(1M) command with the `-e` option and with one or more of the following options: `-U` userID, `-G` groupID, or `-A` adminsetID.
   The following example shows how to retrieve the `quota.group` file to use as a template.

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

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

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

2. Save the file and exit the editor.
3. To apply the changes made in the editor, execute the file using the shell. For example:

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

The `-x` option directs the shell to echo the commands it executes. You can omit the `-x` option if desired.

Checking Quotas

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

How to Check for Exceeded Quotas

1. Become superuser.
2. Display the quotas in effect for mounted file systems.
   - Display user quotas, using the following command:

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

For userID, specify the numeric user ID or user name of the account whose quotas are being examined.
For file, specify a file system for the specified user, group, or admin set. The file argument can also be the name of any file in the file system. Typically, file is the name of the root directory of the file system.
Example 1. The following example retrieves user hm1259 quota statistics in the sam6 file system on the server and displays output indicating that this user is not exceeding the quota.

```
# samquota -U hm1259 /sam6

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

Example 2. The following example retrieves user memil quota statistics in all mounted Sun QFS file systems and displays output indicating that this user is exceeding the quota. Note the plus sign (+) in the Blocks row of the output. In the case of the soft quota limit also being exceeded, a plus sign is also displayed in the Files row.

```
# samquota -U memil

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

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

- Display group quotas, using the following command:

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

For groupID, specify the numeric group ID or the group name for the group of users whose quotas are being examined. For file, specify a file system for the specified group. The file argument can also be the name of any file in the file system. Typically, file is the name of the root directory of the file system.
For example, the following command retrieves user quota statistics for the group turtles in the qfs3 file system:

```
# samquota -G turtles /qfs3
```

- Display admin set quotas, using the following command:

```
# samquota -A <adminsetID> [ <file> ]
```

For adminsetID, specify the numeric admin set ID of the administrator set whose quotas are being examined. For file, specify a file system for the specified admin set. The file argument can also be the name of any file in the file system. Typically, file is the name of the root directory of the file system.
For example, the following command retrieves user quota statistics for the admin set 457 in all mounted file systems:

```
# samquota -A 457
```

Changing and Removing Quotas

You can change quotas to adjust the amount of disk space or number of inodes allocated to users. You can also remove quotas from users or from an entire file system. The following subsections describe how to change and remove quotas.

How to Change the Grace Period

You can use the `samquota`(1M) command to change the soft time limit grace period.

1. Retrieve quota statistics for a user, group, or admin set.
   See How to Check for Exceeded Quotas for instructions. The following example retrieves information about group `sam` and shows that this group is over its soft limit.

```
# samquota -G sam /sam6

                         Online Limits                Total Limits
Type    ID    In Use     Soft     Hard    In Use     Soft     Hard
/sam6
Files  group   101        32     2000     2000        32     2000     2000
Blocks group   101     41888*    40000 60000000     43208 60000000 60000000
Grace period                     1w                            1w
---> Online soft limits under enforcement (since 30s ago)
```

2. Examine the output and determine the new limits.
3. Change the soft time limit grace period. The following example shows the `samquota`(1M) command options to use.

```
# samquota -U <userID> -t <interval> <file>
# samquota -G <groupID> -t <interval> <file>
# samquota -A <adminID> -t <interval> <file>
```

   The arguments for these commands are as follows:

   - userID is the numeric user ID or user name of the user whose quotas are being changed.
   - groupID is the numeric group ID or the group name for the group of users whose quotas are being changed.
   - adminID is the numeric admin set ID of the administrator set whose quotas are being changed.
   - interval is the duration of the grace period. Specify an integer to indicate the quantity and specify a unit of time, if desired. The default unit is `s`, which indicates seconds. You can specify `w` for weeks, `d` for days, `h` for hours, or `m` for minutes.
   - file is the file system for the specified user, group, or admin set. The file argument can also be the name of any file in the file system. Typically, file is the name of the root directory of the file system

Example – Changing the Grace Period

1. To change the grace period for user `memil`, first verify the quotas:

```
# samquota -U memil /sam6

                         Online Limits                Total Limits
Type    ID    In Use     Soft     Hard    In Use     Soft     Hard
/sam6
Files   user 130967         4      500      750         4      500      750
Blocks  user 130967     41016+   40000    50000     41016    50000    50000
Grace period                    3d                            0s
---> Warning:  online soft limits to be enforced in 2d23h59m7s
```

2. Shorten the grace period:

```
# samquota -U memil -t 1d /sam6
```

3. Verify the new quotas.

```
# samquota -U memil /sam6
                         Online Limits                Total Limits
Type    ID    In Use     Soft     Hard    In Use     Soft     Hard
/sam6
Files   user 130967         4      500      750         4      500      750
Blocks  user 130967     41016+   40000    50000     41016    50000    50000
Grace period                    1d                            0s
---> Warning:  online soft limits to be enforced in 23h58m31s
```

Changing the Grace Period Expiration

If a user has exceeded the soft quota limit, changing the grace period itself does not modify the expiration timer of any grace periods that have already started. If the grace period is already in effect, you can use the `samquota`(1M) command to modify the grace period in one of the following ways:

- Clear the grace period timer - The next time the user allocates a file or block while still over a soft limit, the grace period timer is reset and the grace period restarts.
  The following example shows the command used to clear the timer so it starts counting the next time a user in group `sam` attempts to allocate a block or file in `/sam6`.

Example – Clearing the Timer

```
# samquota -G sam -x clear /sam6

Setting Grace Timer:  continue? y

# samquota -G sam  /sam6

                                Online Limits                Total Limits
        Type    ID    In Use     Soft     Hard    In Use     Soft     Hard
/sam6
Files  group   101        32     2000     2000        32     2000     2000
Blocks group   101     41888+   40000 60000000     43208 60000000 60000000
Grace period                   1w                            1w
---> Warning:  online soft limits to be enforced in 6d23h59m56s
```

- Reset the grace period timer - When an expiration period is reset, the timer is reset and the grace period restarts.
  The following example resets the grace period.

Example – Resetting the Grace Period Timer

```
# samquota -G sam -x reset /sam6

Setting Grace Timer:  continue? y

# samquota -G sam  /sam6

                                 Online Limits                Total Limits
        Type    ID    In Use     Soft     Hard    In Use     Soft     Hard
/sam6
Files  group   101        32     2000     2000        32     2000     2000
Blocks group   101     41888    40000 60000000     43208 60000000 60000000
Grace period                   1w                           1w
---> Warning:  online soft limits to be enforced in 6d23h59m52s
```

- Set the grace period to a value - The timer is set to a value, and it starts counting down immediately from that value. There are no restrictions on this value. The value can be larger than the grace period. The following example sets a very long expiration period.

Example – Setting a Very Long Grace Period

```
# samquota -G sam -x 52w /sam6

Setting Grace Timer:  continue? y

# samquota -G sam  /sam6

                                 Online Limits                Total Limits
        Type    ID    In Use     Soft     Hard    In Use     Soft     Hard
/sam6
Files  group   101        32     2000     2000        32     2000     2000
Blocks group   101     41888+   40000 60000000     43208 60000000 60000000
Grace period                   1w                           1w
---> Warning:  online soft limits to be enforced in 51w6d23h59m54s
```

- Expire the grace period timer - The timer is set to expire immediately. The following example expires the grace period.

Example – Expiring the Grace Period Timer

```
# samquota -G sam -x expire /sam6

Setting Grace Timer:  continue? y

# samquota -G sam  /sam6

                                 Online Limits                Total Limits
        Type    ID    In Use     Soft     Hard    In Use     Soft     Hard
/sam6
Files  group   101        32     2000     2000        32     2000     2000
Blocks group   101     41888    40000 60000000     43208 60000000 60000000
Grace period                   1w                           1w
---> Online soft limits under enforcement (since 6s ago)
```

How to Inhibit Additional File System Resource Allocations

When the file system detects that quota values are not consistent for a user, group, or admin set, it prevents that user, group, or admin set from using any more system resources. You can inhibit file system resource allocations by creating inconsistent quota values. For example, you can inhibit further allocation if the hard block or file limits are lower than the soft block or file limits, or if a user's soft limit is larger than the user's hard limit.

The file system treats an inconsistent quota setting as a special quota. You can set inconsistent quota values into record zero of the user, group, or admin set ID quota files, and from there they can become the default values for new users, groups, or admin set IDs.

The following procedure shows how to inhibit further system resource allocations for a user, group, or admin set.

1. Become superuser.
2. Obtain, save, and examine current quota information.
   The following example shows how to retrieve current group quota information for group sam and write it to a backup file.

```
# samquota -G sam -e /sam6 | & tee restore.quota.sam

# Type  ID
#                Online Limits                          Total   Limits
#            soft            hard                   soft            hard
# Files
# Blocks
# Grace Periods
#
samquota -G 101 \
      -f     2000:s:o -f     2000:h:o         -f     2000:s:t -f     2000:h:t \
      -b    40000:s:o -b 60000000:h:o         -b 60000000:s:t -b 60000000:h:t \
                 -t  1w:o                              -t 1w:t \
      -x 51w6d23h59m:o                       -x clear     /sam6
```

   To obtain quota information about a user quota, specify the -U userID option in place of the -G option. To obtain quota information about an admin set quota, specify the -A adminID option in place of the -G option.

3. Use the samquota(1M) command to set soft quotas to nonzero quotas and hard quotas to zero quotas.
   The following command sets the quotas for group sam to be inconsistent:

```
# samquota -G sam -f 1:s -f 0:h -b 1:s -b 0:h /sam6
```

   To make the quotas for users or admin sets inconsistent, specify the -U userID or{{-A}} adminID option in place of the -G option.

4. Use the samquota(1M) command to verify your changes, as in the following example:

```
# samquota -G sam /sam6

Online Limits                Total Limits
Type    ID    In Use     Soft     Hard    In Use     Soft     Hard
/sam6
Files  group   101        32!       1        0        32!       1        0
Blocks group   101     41888!       1        0     43208!       1        0
Grace period                1w                           1w
---> Quota values inconsistent; zero quotas in effect.
```

   In the preceding output, a zero quota is in effect. The exclamation point characters (!) indicate the over-quota condition in the output.

5. Use the sh(1) and samquota(1M) commands to restore the group's quota to what it was before the file/block allocation was inhibited and then to verify the changed quotas. The following example shows these commands.

```
# sh restore.quota.sam
Setting Grace Timer:  continue? y
Setting Grace Timer:  continue? y
# samquota -G sam /sam6
Online Limits                Total Limits
Type    ID    In Use     Soft     Hard    In Use     Soft     Hard
/sam6
Files  group   101        32     2000     2000        32     2000     2000
Blocks group   101     41888+   40000 60000000     43208 60000000 60000000
Grace period                1w                           1w
---> Warning:  online soft limits to be enforced in 6d23h59m54s
```

   To perform this operation on a user quota, specify the -U userID option in place of the -G option. To perform this operation on an admin set quota, specify the -A adminID option in place of the -G option.

How to Remove the Quotas for a File System

To remove or disable quotas for a file system, disable quotas in the mount process.

1. Become superuser.
2. (Optional) Use a text editor to add the `noquota` mount option to the `/etc/vfstab` or `samfs.cmd` file.
   As an alternative, you can specify `noquota` as an option when you issue the `mount` command. See Step 4.
3. If the file system is mounted, use the `umount`(1M) command to unmount the file system. For example:

```
# umount /myfs
```

   If you have difficulty unmounting the file system, see Unmounting a File System.

4. Remount the file system using the `mount`(1M) command.
   If you did not perform Step 2, include the `noquota` option with the `mount`(1M) command. For example:

```
# mount -o noquota /myfs
```

5. Manage the quota files by doing one of the following:
   - If you expect to reinstate the quota feature later and therefore do not want to destroy the quota files, unmount the file system, run the `samfsck`(1M) command with its -F option on the file system, and remount the file system with the `noquota` option removed.
   - If you do not expect to reinstate the quota feature or if you want to reclaim the space consumed by the quota files, use the `rm`(1) command to remove the `.quota_u`, `.quota_g`, and `.quota_a` files. For example:

```
# rm /myfs/.quota_[agu]
```

How to Correct Quotas

1. Become superuser.
2. If the file system is mounted, use the `umount`(1M) command to unmount the file system. For example:

```
# umount /myfs
```

   If you have difficulty unmounting the file system, see Unmounting a File System.

3. Perform a file system check. The `samfsck`(1M) command updates records allocated in the quota files with correct, current usage information. For example:

```
# samfsck -F myfs
```

4. Use the `mount`(1M) command to remount the file system. For example:

```
# mount /myfs
```

Advanced File System Topics

This section discusses advanced topics that are beyond the scope of basic system administration and usage.

Using Daemons, Processes, and Tracing

It is useful to have an understanding of system daemons and processes when you are debugging. This section describes the Sun SAM and Sun QFS daemons and processes. It also provides information about daemon tracing.

Daemons and Processes

All daemons are named in the form `sam-daemon_named`. Processes are named in a similar manner, except that processes do not end in the lowercase letter d.

The following table shows some of the daemons and processes that can run on your system. Others, such as `sam-genericd` and `sam-catserverd`, might also be running, depending on system activities.

Table – Daemons and Processes

Process	Description
sam-amld	Initializes the Sun SAM automated library daemons: `sam-catserverd`, `sam-scannerd`, and `sam-robotsd`.
`sam-archiverd`	Automatically archives Sun SAM files. This process runs as long as the Sun SAM file system is mounted.
`sam-catserverd`	Keeps track of media in Sun SAM and SAM-QFS library catalogs.
`sam-fsd`	Master daemon.
`sam-rftd`	Transfers data between multiple Sun SAM host systems.
`sam-robotsd`	Starts and monitors automated library media changer control daemons.
`sam-scannerd`	Monitors all manually mounted removable media devices. The scanner periodically checks each device for inserted archive media cartridges.
`sam-sharefsd`	Invokes the Sun QFS shared file system daemon.
`sam-releaser`	Attempts to release disk space occupied by previously archived files on Sun SAM file systems until a low-water mark is reached. The releaser is started automatically when a high-water mark is reached on disk cache and stops when it has finished releasing files. This is a process, not a daemon.
`sam-stagealld`	Controls the associative staging of Sun SAM files.
`sam-stagerd`	Controls the staging of Sun SAM files.
`sam-rpcd`	Controls the remote procedure call (RPC) application programming interface (API) server process.
sam-robotsd	Starts and monitors the execution of the media changer library control daemons for Sun SAM.

When you run the software, SMF starts the `sam-fsd` daemon. It should restart automatically in case of failure.

In a shared file system, a `sam-fsd` daemon is always active. In addition, one `sam-sharefsd` daemon is active for each mounted shared file system.

When a shared file system is mounted, the software starts a shared file system daemon (`sam-sharefsd`). TCP sockets are used to communicate between the server and client hosts. All clients that connect to the metadata server are validated against the hosts file.

> Note -
> See the `hosts.fs`(4) man page for more information about the hosts file.

The `sam-sharedfsd` daemon on the metadata server opens a listener socket on the port named `sam-qfs`. During the Sun QFS installation process, the `sam-qfs` entry is automatically added to `/etc/services` file. Do not remove this entry. In addition, the shared file system port is defined in the `/etc/inet/services` file as port number 7105. Verify that this port does not conflict with another service.

Note -
Before the Sun QFS 4U2 release, one port per file system was required. You can remove these entries from your file.

All metadata operations, block allocation and deallocation, and record locking are performed on the metadata server. The sam-sharefsd daemon does not keep any information. Hence, it can be stopped and restarted without causing any consistency problems for the file system.

Trace Files

Several processes can write messages to trace files. These messages contain information about the state and progress of the work performed by the daemons. The messages are primarily used by Sun Microsystems staff to improve performance and diagnose problems. The message content and format are subject to change from release to release.

Trace files can be used in debugging. By default, trace files are not enabled. You can enable trace files by editing the defaults.conf file. You can enable tracing for all processes, or you can enable tracing for individual processes. For information about the processes that you can trace, see the defaults.conf(4) man page.

By default, trace files are written to the /var/opt/SUNWsamfs/trace directory. In that directory, the trace files are named for the processes (archiver, catserver, fsd, ftpd, recycler, sharefsd, and stager). You can change the names of the trace files by specifying directives in the defaults.conf configuration file. You can also set a limit on the size of a trace file and rotate your tracing logs. For information about controlling tracing, see the defaults.conf(4) man page.

Trace File Content

Trace file messages contain the time and source of the message. The messages are produced by events in the processes. You can select the events by using directives in the defaults.conf file.

The default events are as follows:

- Customer notification syslog or notify file messages
- Nonfatal program errors
- Fatal syslog messages
- Process initiation and completion
- Other miscellaneous events

You can also trace the following events:

- Memory allocations
- Interprocess communication
- File actions
- Operator messages
- Queue contents when changed
- Other miscellaneous events

The default message elements (program name, process ID (PID), and time) are always included and cannot be excluded. Optionally, the messages can also contain the following elements:

- The date (The time is always included.)
- The source file name and line number
- The event type

Trace File Rotation

To prevent trace files from growing indefinitely, the sam-fsd daemon monitors the size of the trace files and periodically executes the following command:

```
/opt/SUNWsamfs/sbin/trace_rotate
```

This script moves the trace files to sequentially numbered copies. You can modify this script to suit your operation. Alternatively, you can provide this function using cron(1) or some other facility.

Determining Which Processes Are Being Traced

To determine which processes are being traced currently, enter the sam-fsd(1M) command at the command line. The following example shows the output from this command.

Example – *sam-fsd*(1M) Command Output

```
# sam-fsd
Trace file controls:
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
             cust err fatal misc proc date
             size    10M  age 0
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
             cust err fatal ipc misc proc queue date module
             size    10M  age 0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
             cust err fatal misc proc date
             size    10M  age 0
sam-dbupd     /var/opt/SUNWsamfs/trace/sam-dbupd
             cust err fatal misc proc date
             size    10M  age 0
sam-fsalogd   /var/opt/SUNWsamfs/trace/sam-fsalogd
             cust err fatal misc proc date
             size    10M  age 0
sam-fsd       /var/opt/SUNWsamfs/trace/sam-fsd
             cust err fatal misc proc date
             size    10M  age 0
sam-rftd      /var/opt/SUNWsamfs/trace/sam-rftd
             cust err fatal misc proc date
             size    10M  age 0
sam-recycler  /var/opt/SUNWsamfs/trace/sam-recycler
             cust err fatal ipc misc proc date module type
             size    10M  age 0
sam-nrecycler /var/opt/SUNWsamfs/trace/sam-nrecycler
             cust err fatal misc proc date
             size    10M  age 0
sam-sharefsd  /var/opt/SUNWsamfs/trace/sam-sharefsd
             cust err fatal misc proc date
             size    10M  age 0
sam-stagerd   /var/opt/SUNWsamfs/trace/sam-stagerd
             cust err fatal ipc misc proc date module
             size    10M  age 0
sam-serverd   /var/opt/SUNWsamfs/trace/sam-serverd
             cust err fatal misc proc date
             size    10M  age 0
sam-clientd   /var/opt/SUNWsamfs/trace/sam-clientd
             cust err fatal misc proc date
             size    10M  age 0
fsmgmt        /var/opt/SUNWsamfs/trace/fsmgmt
             cust err fatal misc proc date
             size    10M  age 0
sam-shrink    /var/opt/SUNWsamfs/trace/sam-shrink
             cust err fatal misc proc date
             size    10M  age 0
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
#
```

For more information about enabling trace files, see the defaults.conf(4) man page and the sam-fsd(1M) man page.

Using the **setfa**(1) Command to Set File Attributes

Sun QFS file systems enable end users to set performance attributes for files and directories. Applications can enable these performance features on a per-file or per-directory basis. The following sections describe how the application programmer can use these features to select file attributes for files and directories, to preallocate file space, to specify the allocation method for the file, and to specify the disk stripe width.

For more information about implementing the features described in the following subsections, see the `setfa`(1) man page.

Selecting File Attributes for Files and Directories

The `setfa`(1) command sets attributes on a new or existing file. The file is created if it does not already exist.

You can set attributes on a directory as well as a file. When using `setfa`(1) with a directory, files and directories created within that directory inherit the attributes set in the original directory. To reset attributes on a file or directory to the default, use the{{– d}} (default) option. When the{{– d}} option is used, attributes are first reset to the default and then other attributes are processed.

Preallocating File Space

An end user can preallocate space for a file. This space is associated with a file so that no other files in the file system can use the disk addresses allocated to this file. Preallocation ensures that space is available for a given file, which avoids a file-system-full condition. Preallocation is assigned at the time of the request rather than when the data is actually written to disk.

Note that space can be wasted by preallocation of files. If the file size is less than the allocation amount, the kernel allocates space to the file from the current file size up to the allocation amount. When the file is closed, space below the allocation amount is not freed.

You can preallocate space for a file by using the `setfa`(1) command with either the `-L` or the{{– l}} (lowercase letter L) option. Both options accept a file length as their argument. Use the `-L` option for an existing file, which can be empty or contain data. Use the `-l` option for a file that has no data yet. If you use the `-l` option, the file cannot grow beyond its preallocated limit.

For example, to preallocate a 1-gigabyte file named `/qfs/file_alloc`, type the following:

```
# setfa -l 1g /qfs/file_alloc
```

After space for a file has been preallocated, truncating a file to 0 length or removing the file returns all space allocated for a file. There is no way to return only part of a file's preallocated space to the file system. In addition, if a file is preallocated with the `-l` option, there is no way to extend the file beyond its preallocated size in future operations.

Selecting a File Allocation Method and Stripe Width

By default, a file uses the allocation method and stripe width specified at mount time (see the `mount_samfs`(1M) man page). However, an end user might want to use a different allocation scheme for a file or directory. The user could do this by using the `setfa`(1) command with the `-s` (stripe) option.

The allocation method can be either round-robin or striped. The `-s` option specifies the allocation method and the stripe width, as shown in the following table.

Table – File Allocations and Stripe Widths

`-s` Option	Allocation Method	Stripe Width	Explanation
0	Round-robin	Not applicable	The file is allocated on one device until that device has no space.
1–255	Striped	1–255 DAUs	The file is striped across all disk devices with this number of DAUs per disk.

The following example shows how to create a file explicitly by specifying a round-robin allocation method:

```
# setfa -s 0 /qfs/100MB.rrobin
```

The following example shows how to create a file explicitly by specifying a striped allocation method with a stripe width of 64 DAUs (preallocation is not used):

```
# setfa -s 64 /qfs/file.stripe
```

Selecting a Striped Group Device

Striped group devices are supported for Sun QFS file systems only.

A user can specify that a file begin allocation on a particular striped group. If the file allocation method is round-robin, the file is allocated on the designated stripe group.

The following example shows `setfa`(1) commands specifying that `file1` and `file2` be independently spread across two different striped groups.

Example – ***setfa***(1) Commands to Spread Files Across Striped Groups

```
# setfa -g0 -s0 file1
# setfa -g1 -s0 file2
```

This capability is particularly important for applications that must achieve levels of performance that approach raw device speeds. For more information, see the `setfa`(1) man page.

Accommodating Large Files

When manipulating very large files, pay careful attention to the size of disk cache that is available on the system. If you try to write a file that is larger than your disk cache, behavior differs depending on the type of file system that you are using:

- If you are using a non-archiving file system, the system returns an ENOSPC error.
- If you are using an archiving file system, the program blocks, waiting for space that might never exist, because the available disk space is insufficient to handle the request.

If you are operating within an archiving environment and your application must write a file that is larger than the disk cache, you can segment the file with the `segment`(1) command. For more information about the `segment`(1) command, see the `segment`(1) man page or see Using Segmented Files.

Configuring a Multireader File System

The multireader file system consists of a single writer host and multiple reader hosts. The `writer` and `reader` mount options that enable the multireader file system are compatible with Sun QFS file systems only. The mount options are described in this section and on the `mount_samfs`(1M) man page.

You can mount the multireader file system on the single writer host by specifying the{{– o writer}} option with the `mount`(1M) command. The host system with the `writer` mount option is the only host system that is allowed to write to the file system. The `writer` host system updates the file system. You must ensure that only one host in a multireader file system has the file system mounted with the `writer` mount option enabled. If{{– o writer}} is specified, directories are written through to disk at each change and files are written through to disk at close.

> Caution -
> The multireader file system can become corrupted if more than one writer host has the file system mounted at one time. It is the site administrator's responsibility to ensure that this situation does not occur.

You can mount a multireader file system on one or more reader hosts by specifying the{{– o reader}} option with the `mount`(1M) command. There is no limit to the number of host systems that can have the multireader file system mounted as a reader.

A major difference between the multireader file system and a Sun QFS shared file system is that the multireader host reads metadata from the disk, and the client hosts of a Sun QFS shared file system read metadata over the network. The Sun QFS shared file system supports multireader hosts. In this configuration, multiple shared hosts can be adding content while multiple reader hosts are distributing content.

> **Note -**
> You cannot specify the `writer` option on any host if you are mounting the file system as a Sun QFS shared file system. You can, however, specify the `reader` option. If you want a Sun QFS shared file system client host to be a read-only host, mount the Sun QFS shared file system on that host with the `reader` mount option. In addition, set the `sync_meta` mount option to `1` if you use the `reader` option in a Sun QFS shared file system. For more information about the Sun QFS shared file system, see Configuring a Shared File System . For more information about mount options, see the `mount_samfs`(1M) man page.

You must ensure that all readers in a multireader file system have access to the device definitions that describe the ma device. Copy the lines from the `mcf` file that resides on the primary metadata server host to the `mcf` files on the alternate metadata servers. After copying the lines, you might need to update the information about the disk controllers because, depending on your configuration, disk partitions might not show up the same way across all hosts.

In a multireader file system environment, the Sun QFS software ensures that all servers accessing the same file system can always access the current environment. When the writer closes a file, the Sun QFS file system immediately writes all information for that file to disk. A `reader` host can access a file after the file is closed by the writer. You can specify the `refresh_at_eof` mount option to help ensure that no host system in a multireader file system gets out of sync with the file system.

By default, the metadata information for a file on a `reader` host is invalidated and refreshed every time a file is accessed. If the data changed, it is invalidated. This includes any type of access, whether through `cat`(1), `ls`(1), `touch`(1), `open`(2), or other methods. This immediate refresh rate ensures that the data is correct at the time the refresh is done, but it can affect performance. Depending on your site preferences, you can use the `mount`(1M) command's{{– o invalid=}}n option to specify a refresh rate between 0 seconds and 60 seconds. If the refresh rate is set to a small value, the Sun QFS file system reads the directory and other metadata information n seconds after the last refresh. More frequent refreshes result in more overhead for the system, but stale information can exist if n is nonzero.

> **Caution -**
> If a file is open for a read on a `reader` host, there is no protection against that file being removed or truncated by the writer. You must use another mechanism, such as application locking, to protect the reader from inadvertent writer actions.

Understanding I/O Types

The Sun QFS file systems support paged I/O, direct I/O, and switching between the I/O types. The following sections describe these I/O types.

Paged I/O

When paged I/O is used, user data is cached in virtual memory pages, and the kernel writes the data to disk. The standard Solaris OS interfaces manage paged I/O. Paged I/O (also called buffered or cached I/O) is selected by default.

Direct I/O

Direct I/O is a process by which data is transferred directly between the user's buffer and the disk. This means that much less time is spent in the system. For performance purposes, specify direct I/O only for large, block-aligned, sequential I/O.

The `setfa`(1) command and the `sam_setfa`(3) library routine both have a{{– D}} option that sets the direct I/O attribute for a file or directory. If applied to a directory, files and directories created in that directory inherit the direct I/O attribute. After the{{– D}} option is set, the file uses direct I/O.

You can also select direct I/O for a file by using the Solaris OS `directio`(3C) function call. If you use the function call to enable direct I/O, the setting lasts only while the file is active.

To enable direct I/O on a file system basis, do one of the following:

- Specify the{{– o forcedirectio}} option with the `mount`(1M) command.
- Put the `forcedirectio` keyword in the mount option column of the `/etc/vfstab` file, or use it as a directive in the `samfs.cmd` file.

For more information, see the `setfa`(1), `sam_setfa`(3), `directio`(3C), `samfs.cmd`(4), and `mount_samfs`(1M) man pages.

I/O Switching

By default, paged I/O is performed, and I/O switching is disabled. However, the Sun QFS file systems support automatic I/O switching, a process by which a site-defined amount of paged I/O occurs before the system switches automatically to direct I/O.

I/O switching should reduce page cache usage on large I/O operations. To enable I/O switching, use samu(1M), or use the dio_wr_consec and dio_rd_consec parameters as directives in the samfs.cmd file or as options with the mount(1M) command.

For more information about these options, see the mount_samfs(1M) or samfs.cmd(4) man pages.

Configuring WORM-FS File Systems

About WORM-FS File Systems

Write once read many (WORM) technology is used in many applications for data integrity reasons and because of the accepted legal admissibility of stored files that use the technology.

> Note -
> The WORM-FS package (SUNWsamfswm) is included with the Sun QFS software packages, but must be installed separately by using the pkgadd command.

The WORM-FS feature offers default and customizable file-retention periods, data and path immutability, and subdirectory inheritance of the WORM setting.

WORM-FS can operate in one of two modes:

- Sun standard compliance mode (referred to as standard mode), which is the default
- Sun emulation compliance mode (referred to as emulation mode), which is designed to provide compatibility with the emulation mode of the Sun StorageTek 5320 network attached storage (NAS) appliance and is similar to an interface defined by Network Appliance

One difference between standard and emulation mode is a restriction on the nature of files that can be retained. Specifically, in standard mode, files with any UNIX executable permissions cannot be retained. There is no such restriction in emulation mode. The restriction in standard mode exists because of the nature of the retention trigger defined for NFS and FTP. For these protocols, retention is requested by specifying that the setuid mode be set on the file. Once a file is retained, a client will see the setuid mode bit set, but the restriction on executable files will prevent the possible security hole of allowing an executable file owned by the root user to be made WORM and therefore impossible to remove. A benefit of this approach is that the user or application can more easily determine which files on the system are indeed WORM-protected files.

Using WORM-FS With NFS Clients

If you are using WORM-FS on Solaris 10 or later with NFS clients connected to it, ensure that NFS version 4 is enabled on the NFS clients and sever.

If you are running an older version of the Solaris OS (before Solaris 10) and NFS version 3, the NFS client might not show the WORM-FS files. In this case, follow these steps:

1. Add the following line to the /etc/system file:

```
set nfs:nfs_allow_preepoch_time = 1
```

2. Reboot the system.

Enabling the WORM-FS Feature

There are four mount options that can be used to enable the WORM-FS feature:

The following table shows four mount options that you can use to enable the WORM-FS feature.

Option	Brief Description	Notes

`worm_capable`	Standard WORM mode	The WORM trigger command, `chmod 4000` file-name/directory-name, is used to set the WORM bit on a file or directory.
`worm_lite`	Relaxes some of the standard WORM mode restrictions	The system administrator is allowed to delete files before retention expiration and reduce the file retention period. File data and path integrity remain immutable. See WORM "Lite" Options for more information.
`worm_emul`	WORM emulation mode, which is designed to provide compatibility with the emulation mode of the Sun StorageTek 5320 network attached storage (NAS) appliance	This provides standard WORM functionality with a different WORM trigger. The WORM bit is set by changing a directory or file from writeable to read-only.
`emul_lite`	WORM emulation mode, which is designed to provide compatibility with the “lite” version of the Sun StorageTek 5320 network attached storage (NAS) appliance	This provides standard WORM lite functionality with a different WORM trigger. The WORM bit is set by changing a directory or file from writeable to read-only. As with the worm_lite option, the administrator can carry out special operations on files. See WORM "Lite" Options for more information.

These four mount options are somewhat exclusive. You can upgrade from “lite” to standard WORM mode, but you cannot change from standard WORM mode to emulation mode, or from emulation to standard mode. These options can be provided on the command line when the file system is mounted, listed in `/etc/vfstab`, or provided in `/opt/SUNWsamfs/samfs.cmd`. The normal rules of precedence for mount options apply.

The WORM attribute is stored in the mount table and enables WORM files to be created in directories anywhere in the file system.

> Note -
> You must have system administration privileges to set a WORM mount option in `/etc/vfstab`.

The following example shows an example of WORM-FS mount options. The file system `samfs1` mounted at `/samfs1` is WORM-capable and has the default retention period for files set to 60 minutes.

Example – Using WORM-FS Mount Options

```
# cat /etc/vfstab
  #device device mount FS fsck mount mount
  #to mount to fsck point type pass at boot options
  #
  fd - /dev/fd fd - no -
  /proc - /proc proc - no -
  /dev/dsk/c0t0d0s1 - - swap - no -
  samfs1 - /samfs1 samfs - yes worm_capable,def_retention=60
  swap - /tmp tmpfs - yes -
```

After the WORM-FS feature has been enabled and at least one WORM file is resident in the file system, the file system's superblock is updated to reflect the WORM capability. Any subsequent attempt to rebuild the file system through `sammkfs` will fail, unless you are using the `worm_lite` or `emul_lite` mount option.

WORM “Lite” Options

The `worm_lite` and `emul_lite` mount options create a modified WORM environment that eases the restrictions on actions that can be taken on WORM-enabled volumes and retained files. The WORM lite options can be a solution for companies with document management and retention policies requiring data retention guarantees but not the strict constraints that WORM places on systems. Mechanisms exist to alter and even reverse some data retention decisions.

The WORM lite options can also be used for testing and configuring WORM systems and applications before upgrading to the more strict standard WORM policies.

The WORM lite environment behaves similarly to the standard WORM mode. File data and path remain immutable, but the system administrator is allowed to carry out the following special actions:

- Remove WORM files before the retention time has expired
- Shorten the retention time on WORM files
- Delete WORM Lite-enabled volumes or rebuild them using the `sammkfs` command

Creating WORM Files

A WORM mount option enables a file system to contain WORM files, but it does not automatically create WORM files. To create a WORM file, you must first make the directory WORM-capable. To do this, create an ordinary directory and then use a WORM trigger command to set the WORM bit on the directory. Depending on the mount option being used, the following WORM trigger commands are available:

- Use `chmod 4000` directory-name to set the WORM bit if you are using the `worm_capable` or `worm_lite` mount option.
- Remove the write permissions on the directory to set the WORM bit if you are using the `worm_emul` or `emul_lite` mount option.

After setting the WORM bit on a directory, you can create files in that directory and then use the appropriate WORM trigger to set the WORM bit on files that you want retained. The WORM trigger is the same for both files and directories.

The following are examples of using the WORM trigger for each of the four mount options using the system-wide default retention value:

Example 1. WORM trigger is ***chmod 4000***

Simple application of the WORM trigger using standard WORM functionality:

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -       /samfs1  samfs   -       no     bg,worm_capable

[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod 4000 WORM
[root@ns-east-44]# sls -D

WORM:
  mode: drwxr-xr-x  links:   2  owner: root      group: root
  length:      4096  admin id:      0  inode:     1025.1
  access:      Jan 30 15:50  modification: Jan 30 15:50
  changed:     Jan 30 15:50  attributes:   Jan  1  1970
  creation:    Jan 30 15:50  residence:    Jan 30 15:50
  worm-capable        retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod 4000 test
[root@ns-east-44]# sls -D

test:
  mode: -r-Sr--r--  links:   1  owner: root      group: root
  length:         0  admin id:      0  inode:     1026.3
  access:      Jan 30 15:51  modification: Jan 30 15:51
  changed:     Jan 30 15:51  retention-end: Mar  1 15:51 2007
  creation:    Jan 30 15:51  residence:    Jan 30 15:51
  retention:   active        retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
rm: test not removed: Read-only file system
[root@ns-east-44]# ls
test
```

Example 2. WORM trigger is **chmod 4000**

Simple application of the WORM trigger using standard WORM lite functionality:

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -       /samfs1  samfs   -       no     bg,worm_lite

[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod 4000 WORM
[root@ns-east-44]# sls -D

WORM:
  mode: drwxr-xr-x  links:   2  owner: root      group: root
  length:      4096  admin id:      0  inode:     1025.1
  access:      Jan 30 16:12  modification: Jan 30 16:12
  changed:     Jan 30 16:12  attributes:   Jan  1  1970
  creation:    Jan 30 16:12  residence:    Jan 30 16:12
  worm-capable        retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod 4000 test
[root@ns-east-44]# sls -D
test:
  mode: -r-Sr--r--  links:   1  owner: root      group: root
  length:         0  admin id:      0  inode:     1026.1
  access:      Jan 30 16:13  modification: Jan 30 16:13
  changed:     Jan 30 16:13  retention-end: Mar  1 16:13 2007
  creation:    Jan 30 16:13  residence:    Jan 30 16:13
  retention:   active        retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
[root@ns-east-44]# ls
[root@ns-east-44]#
```

Example 3. WORM trigger is **`chmod -w`**

Simple application of the WORM trigger using WORM emulation mode:

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -       /samfs1  samfs   -       no     bg,worm_emul

[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod -w WORM
[root@ns-east-44]# sls -D

WORM:
  mode: drwxr-xr-x  links:   2  owner: root      group: root
  length:      4096  admin id:      0  inode:     1025.1
  access:      Jan 30 16:26  modification: Jan 30 16:26
  changed:     Jan 30 16:26  attributes:   Jan  1  1970
  creation:    Jan 30 16:26  residence:    Jan 30 16:26
  worm-capable        retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod -w test
[root@ns-east-44]# sls -D

test:
  mode: -r--r--r--  links:   1  owner: root      group: root
  length:         0  admin id:      0  inode:     1026.1
  access:      Jan 30 16:26  modification: Jan 30 16:26
  changed:     Jan 30 16:26  retention-end: Mar  1 16:26 2007
  creation:    Jan 30 16:26  residence:    Jan 30 16:26
  retention:   active        retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
rm: test not removed: Read-only file system
[root@ns-east-44]# ls
test
```

Example 4. WORM trigger is **`chmod -w`**

Simple application of the WORM trigger using WORM emulation lite mode:

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -       /samfs1  samfs   -       no     bg,emul_lite

[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod -w WORM
[root@ns-east-44]# sls -D

WORM:
  mode: drwxr-xr-x  links:   2  owner: root      group: root
  length:      4096  admin id:      0  inode:     1025.1
  access:      Jan 30 16:36  modification: Jan 30 16:36
  changed:     Jan 30 16:36  attributes:   Jan  1  1970
  creation:    Jan 30 16:36  residence:    Jan 30 16:36
  worm-capable        retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod -w test
[root@ns-east-44]# sls -D

test:
  mode: -r--r--r--  links:   1  owner: root      group: root
  length:         0  admin id:      0  inode:     1026.1
  access:      Jan 30 16:36  modification: Jan 30 16:36
  changed:     Jan 30 16:36  retention-end: Mar  1 16:36 2007
  creation:    Jan 30 16:36  residence:    Jan 30 16:36
  retention:   active        retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
[root@ns-east-44]# ls
[root@ns-east-44]#
```

Note -
Use care when applying the WORM trigger. The file data and path cannot be changed after the file has the WORM feature applied. Once this feature is applied to a file, it is irrevocable. Further, once the WORM trigger is applied to a file, its volume also become a WORM volume and remains that way. The volume can only be destroyed using a volume management or RAID interface. If one of the WORM "lite" options was used to create it, the volume can also be rebuilt by using `sammkfs`.

Retention Periods

The WORM-FS feature also includes file-retention periods that can be customized. Assigning a retention period to a file maintains the WORM features in that file for the specified period of time.

Note -
Retention periods cannot extend beyond 01/18/2038 when initially assigning or extending the period using Solaris/UNIX utilities. This is due to the fact these utilities use signed 32 bit numbers to represent time in seconds. Time is measured from the epoch which is January 1, 1970. 2**31 seconds from the epoch extends to 01/18/2038 around 10:14 PM. You can, however, exceed this date using a default retention period. See Setting the Default Retention Period.

Do one of the following to set a retention period for a file:

- Advance the file’s access time using the `touch` utility, or with a program using the `libc` subroutine `utimes()`. The file’s retention period is stored in minutes. After the access time is advanced, use the appropriate WORM trigger to set the WORM bit.
- Use the default retention period for a file. This is accomplished by applying the appropriate WORM trigger and allowing the file system to apply the default retention period. See Setting the Default Retention Period for more information.

The following example shows the creation of a file in a WORM-capable directory, using the WORM trigger on the file (with the `chmod 4000` command), and using the `sls` command to display the file's WORM features. This example uses the default retention period of the file system (60 minutes, as set in Example – Using WORM-FS Mount Options).

Example – Creation of a WORM-Capable Directory and WORM File

```
# cd WORM
# echo "This is a test file" >> test
# sls -D
test:
            mode: -rw-r--r-- links: 1 owner: root group: other
            length: 20 admin id: 0 inode: 1027.1
            access: Oct 30 02:50 modification: Oct 30 02:50
            changed: Oct 30 02:50 attributes: Oct 30 02:50
            creation: Oct 30 02:50 residence: Oct 30 02:50

            checksum: gen no_use not_val algo: 0

# chmod 4000 test
# sls -D
test:
            mode: -r--r--r-- links: 1 owner: root group: other
            length: 20 admin id: 0 inode: 1027.1
            access: Oct 30 02:50 modification: Oct 30 02:50
            changed: Oct 30 02:50 retention-end: Oct 30 2005 03:50
            creation: Oct 30 02:50 residence: Oct 30 02:50
            retention: active retention-period: 0y, 0d, 1h, 0m
            checksum: gen no_use not_val algo: 0
```

With the addition of the WORM-FS feature, three states are possible for a file in a Sun QFS file system:

- Normal
- Retained
- Expired

The normal state represents the state of an ordinary file in a Sun QFS file system. A transition to the retained, or active, state occurs when the WORM bit is set on a file. The expired, or over, state occurs when the file's retention period is exceeded.

When a retention period is assigned to a file and the WORM trigger is applied to it, the file's path and data are immutable. When the retention period expires, the state is changed to "expired" but the path and data remain immutable.

When a file is in an expired state, only two operations are available:

- Extension of the retention period (The retention period cannot be shortened unless you are using a WORM "lite" option.)
- Deletion of the file

If the retention period is extended, the file's state returns to "active" and the new end date and duration are set accordingly.

Both hard and soft links to files can be used with the WORM-FS feature. Hard links can be established only with files that reside in a WORM-capable directory. After a hard link is created, it has the same WORM characteristics as the original file. Soft links can also be established, but a soft link cannot use the WORM features. Soft links to WORM files can be created in any directory in a Sun QFS file system.

Another attribute of the WORM-FS feature is directory inheritance. New directories created under a directory that includes a WORM attribute inherit this attribute from their parent. If a directory has a default retention period set, this retention period is also inherited by any new subdirectories. The WORM bit can be set on any file whose parent directory is WORM-capable. Ordinary users can set the WORM feature on directories and files that they own or have access to by using normal UNIX permissions.

> Note -
> A WORM-capable directory can only be deleted if it contains no WORM files.

Setting the Default Retention Period

The default retention period for a file system can be set as a mount option in the `/etc/vfstab` file. For example:

```
samfs1 - /samfs1 samfs - nobg,worm_emul,def_retention=1y60d
```

The format for setting the default retention period is MyNdOhPm, in which M, N, O, and P are non-negative integers and `y`, `d`, `h`, and `m` stand for years, days, hours, and minutes, respectively. Any combination of these units can be used. For example, `1y5d4h3m` indicates 1 year, 5 days, 4 hours, and 3 minutes; `30d8h` indicates 30 days and 8 hours; and `300m` indicates 300 minutes. This format is backward compatible with software versions prior to 4U5, in which the retention period was specified in minutes. It is important to note, although the granularity of the period is in minutes, the accuracy of the period is based on one day. Also, the function handling days, hours, and minutes does not account for leap years when determining retention periods. You must consider this when using one (or all) of these to set the default retention period.

You can also use the default retention period to set a file or directory's retention period beyond the year 2038. To do this, set the default retention period to a value which exceeds 2038 and mount the file system. Then use the appropriate WORM trigger to apply the default retention period. Here is an example of using the default retention period to set a retention period on a directory and file which exceeds the year 2038.

Example – Extending the Retention Period Beyond 2038

```
[root@ns-east-44]# grep samfs1 /etc/vfstab
samfs1 - /samfs1 samfs - no
bg,worm_capable,def_retention=34y
    [root@ns-east-44]# mount samfs1
    [root@ns-east-44]# cd /samfs1
    [root@ns-east-44]# mkdir WORM
    [root@ns-east-44]# chmod 4000 WORM
    [root@ns-east-44]# sls -D
WORM:
        mode: drwxr-xr-x links: 2 owner: root group: root
        length: 4096 admin id: 0 inode: 1026.1
        access: Feb 20 14:24 modification: Feb 20 14:24
        changed: Feb 20 14:24 attributes: Jul 26 1970
        creation: Feb 20 14:24 residence: Feb 20 14:24
        worm-capable retention-period: 34y, 0d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod 4000 test
[root@ns-east-44]# sls -D
test:
        mode: -r-Sr--r-- links: 1 owner: root group: root
        length: 0 admin id: 0 inode: 1027.1
        access: Feb 20 14:24 modification: Feb 20 14:25
        changed: Feb 20 14:25 retention-end: Feb 20 14:25 2041
            creation: Feb 20 14:24 residence: Feb 20 14:24
    retention: active retention-period: 34y, 0d, 0h, 0m
```

You can also set a default retention period for a directory using the `touch` utility, as described in the following section, Setting the Retention Period Using `touch`. This retention period overrides the default retention period for the file system and is inherited by any subdirectories.

Setting the Retention Period Using **`touch`**

You can use the `touch` utility to set or extend a file's or directory's retention period. You can also use `touch` to shorten the default retention period for a directory (but not for a file).

To set the retention period, you must first advance the file's or directory's access time using `touch`, and then apply the WORM trigger by using the `chmod` command or removing write permissions (depending on the WORM mode in place at the time).

The following example shows the use of the `touch` utility to set a file's retention period, followed by the application of the WORM trigger.

Example – Using ***`touch`*** and ***`chmod`*** to Set the Retention Period

```
# touch -a -t200508181125 test
# sls -D
test:
        mode: -rw-r--r-- links: 1 owner: root group: root
        length: 0 admin id: 0 inode: 1027.1
        access: Aug 18 2005 modification: Aug 18 11:19
        changed: Aug 18 11:19 attributes: Aug 18 11:19
        creation: Aug 18 11:19 residence: Aug 18 11:19

# chmod 4000 test
# sls -D
test:
        mode: -r-Sr--r-- links: 1 owner: root group: root
        length: 0 admin id: 0 inode: 1027.1
        access: Aug 18 2005 modification: Aug 18 11:19
        changed: Aug 18 11:19 retention-end: Aug 18 2005 11:25
        creation: Aug 18 11:19 residence: Aug 18 11:19
        retention: active retention-period: 0y, 0d, 0h, 6m
```

The `-a` option for `touch` is used to change the access time of the file or directory. The `-t` option specifies what time is to be used for the access time field. The format for the time argument is `[[CC]YY]MMDDhhmm[.SS]`, as follows:

- `[CC]` – The first two digits of the year.
- `[YY]` – The second two digits of the year.
- `MM` – The month of the year (01–12).
- `DD` – The day of the month (01–31).
- `hh` – The hour of the day (00–23).
- `mm` – The minute of the hour (00–59).
- `[SS]`– The second of the minute (00–61).

The CC, YY, and SS fields are optional. If CC and YY are not given, the default is the current year. See the `touch` manpage for more information on these options.

To set the retention period to permanent retention, set the access time to its largest possible value: `203801182214.07`.

Extending a File's Retention Period

This example shows an example of using `touch` to extend a file's retention period.

Example – Using *touch* to Extend a File's Retention Period

```
# sls -D test
test:
        mode: -r-Sr--r-- links: 1 owner: root group: root
        length: 0 admin id: 0 inode: 1029.1
        access: Aug 18 11:35 modification: Aug 18 11:33
        changed: Aug 18 11:33 retention-end: Aug 18 2005 11:35
        creation: Aug 18 11:33 residence: Aug 18 11:33
        retention: over retention-period: 0y, 0d, 0h, 2m
# touch -a -t200508181159 test
# sls -D
test:
        mode: -r-Sr--r-- links: 1 owner: root group: root
        length: 0 admin id: 0 inode: 1029.1
        access: Aug 18 11:35 modification: Aug 18 11:33
        changed: Aug 18 11:33 retention-end: Aug 18 2005 11:59
        creation: Aug 18 11:33 residence: Aug 18 11:33
        retention: active retention-period: 0y, 0d, 0h, 26m
```

In this example the retention period was extended to Aug 18, 2005 at 11:59AM, which is 26 minutes from the time the WORM trigger was initially applied.

Note -
Using touch to extend the retention period is independent of the active WORM mode.

Using **sls** to View WORM-FS Files

Use the sls command to view WORM file attributes. The -D option shows whether a directory is WORM-capable. Use this option on a file to display when the retention period began, when it will end, the current retention state, and the duration as specified on the command line.

The retention period start time and duration (in minutes) are stored in the file's inode.
To access this information directly one must use a program similar to the following example:

Example – Program for Direct Access to Retention Period Start Time and Duration

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
/*
* SAMFS includes in /opt/SUNWsamfs/include
*/
#include "stat.h"
#include "lib.h"

/*
* SAMFS libraries in /opt/SUNWsamfs/lib
*/
int
main(int argc, char **argv)
{
   char ibuf[1000];
   struct sam_stat buf;

   if (argc != 2) {
       printf("usage: sam_worm_stat filename\n");
       exit(-1);
   }

   if (sam_stat(argv[1], &buf, sizeof(buf)) == 0) {
       (void)time_string(buf.rperiod_start_time,
buf.rperiod_start_time, ibuf);
       printf("retention period start is %s\n", ibuf);
       printf("retent period duration is %ld minutes\n",
buf.rperiod_duration);
   } else {
       printf("can not sam_worm_stat %s\n", argv[1]);
   }
}
```

The following example shows how sls -D displays a file's retention status.

Example – Using ***sls*** to Find a File's Retention Status

```
# sls -D test
test:
        mode: -r-Sr--r-- links: 1 owner: root group: root
        length: 5 admin id: 0 inode: 1027.1
        access: Aug 18 2008 modification: Aug 18 11:19
        changed: Aug 18 11:19 retention-end: Aug 18 2008 11:25
        creation: Aug 18 11:19 residence: Aug 18 11:19
        retention: active retention-period: 0y, 0d, 0h, 6m
```

In this example, the retention state is active, as shown by the retention: active designation, meaning that the file has the WORM bit set. The retention period started on August 18, 2008, at 11:19 and will end on August 18, 2008, at 11:25. The retention period was specified to be 0 years, 0 days, 0 hours, and 6 minutes.

Using **`sfind`** to Find WORM-FS Files

Use the `sfind` utility to search for files that have certain retention periods. See the `sfind`(1) man page for more information on the options. The following options are available:

- `-ractive` – Finds files whose retention period is active.
- `-rover` – Finds files whose retention periods have expired.
- `-rafter` date – Finds files whose retention period will end after the specified date. The date is specified as `YYYYMMDDHHmm`, where `YYYY` is the year, `MM` is the month, `DD` is the day, `HH` is the hour, and `mm` is minutes.
  The following example shows the command to find files whose retention period expires after 12/24/2004 at 15:00.

  ```
  # sfind -rafter 200412241500
  ```

- `-rremain` time – Finds files that have retention periods with at least the specified amount of time left. The time is specified as `MyNdOhPm`, where `M`, `N`, `O`, and `P` are arbitrary non-negative integers and `y`, `d`, `h`, and `m` represent the number of years, days, hours, and minutes, respectively.
  For example, the following command finds files for which more than 1 year, 10 days, 5 hours, and 10 minutes remain before expiration.

  ```
  # sfind -rremain 1y10d5h10m
  ```

- `-rlonger` time – Finds files that have retention periods longer than the specified amount of time. The time is specified as `MyNdOhPm`, where `M`, `N`, `O`, and `P` are arbitrary non-negative integers and `y`, `d`, `h`, and `m` represent the number of years, days, hours, and minutes, respectively.
  For example, the following command finds files that have retention periods longer than 10 days.

  ```
  # sfind -rlonger 10d
  ```

- `-rpermanent` – Finds files whose retention period is permanent.

Tunable Parameters

This section describes various ways that you can improve file system and archiving performance.

> Note -
> Sun recommends that you experiment with performance tuning outside of a production environment. Tuning these variables incorrectly can have unexpected effects on the overall system.
> If your site has a Sun™ Enterprise Services (SES) support contract, please inform SES if you change performance tuning parameters.

Increasing File Transfer Performance for Large Files

Sun QFS file systems are tuned to work with a mix of file sizes. You can increase the performance of disk file transfers for large files by enabling file system settings.

How to Increase File Transfer Performance

1. Set the maximum device read/write directive.
   The `maxphys` parameter in the Solaris `/etc/system` file controls the maximum number of bytes that a device driver reads or writes

at any one time. The default value for the maxphys parameter can differ, depending on the level of your Sun Solaris OS, but it is typically around 128 kilobytes.
Add the following line to /etc/system to set maxphys to 1 megabytes:

```
set maxphys = 0x100000
```

> Note -
> The maxphys value must be set to a power of two.

2. Set the SCSI disk maximum transfer parameter.
   The sd driver enables large transfers for a specific file by looking for the sd_max_xfer_size definition in the /kernel/drv/sd.conf file. If this definition does not exist, the driver uses the value defined in the sd device driver definition, sd_max_xfer_size, which is 1024 x 1024 bytes.
   To enable and encourage large transfers, add the following line at the end of the /kernel/drv/sd.conf file:

   ```
   sd_max_xfer_size=0x800000;
   ```

3. Set the fibre disk maximum transfer parameter.
   The ssd driver enables large transfers for a specific file by looking for the ssd_max_xfer_size definition in the /kernel/drv/ssd.conf file. If this definition does not exist, the driver uses the value defined in the ssd device driver definition, ssd_max_xfer_size, which is 1024 x 1024 bytes.
   Add the following line at the end of the /kernel/drv/ssd.conf file:

   ```
   ssd_max_xfer_size=0x800000;
   ```

   > Note -
   > On Solaris 10 x86 platforms, this change is made in the /kernel/drv/sd.conf file. For a maximum transfer size of 8 MBytes, the following line is added.sd_max_xfer_size=0x800000

4. Reboot the system.
5. Set the writebehind parameter.
   This step affects paged I/O only.
   The writebehind parameter specifies the number of bytes that are written behind by the file system when paged I/O is being performed on a Sun QFS file system. Matching the writebehind value to a multiple of the RAID's read-modify-write value can increase performance.
   This parameter is specified in units of kilobytes and is truncated to an 8-kilobyte multiple. If set, this parameter is ignored when direct I/O is performed. The default writebehind value is 512 kilobytes. This value favors large-block, sequential I/O.
   Set the writebehind size to a multiple of the RAID 5 stripe size for both hardware and software RAID-5. The RAID-5 stripe size is the number of data disks multiplied by the configured stripe width.
   For example, assume that you configure a RAID-5 device with three data disks plus one parity disk (3+1) with a stripe width of 16 kilobytes. The writebehind value should be 48 kilobytes, 96 kilobytes, or some other multiple, to avoid the overhead of the read-modify-write RAID-5 parity generation.
   For Sun QFS file systems, the DAU (sammkfs(1M) -a command) should also be a multiple of the RAID-5 stripe size. This allocation ensures that the blocks are contiguous.
   You should test the system performance after resetting the writebehind size. The following example shows testing timings of disk writes:

   ```
   # timex dd if=/dev/zero of=/sam/myfile bs=256k count=2048
   ```

   You can set the writebehind parameter from a mount option, from within the samfs.cmd file, from within the /etc/vfstab file, or from a command within the samu(1M) utility. For information about enabling this from a mount option, see the{{– o writebehind=}} n option on the mount_samfs(1M) man page. For information about enabling this from the samfs.cmd file, see the samfs.cmd(4) man page. For information about enabling this from within samu(1M), see the samu(1M) man page.

6. Set the readahead parameter.

This step affects paged I/O only.
The readahead parameter specifies the number of bytes that are read ahead by the file system when paged I/O is being performed on a Sun QFS file system. This parameter is specified in units of kilobytes and is truncated to an 8-kilobyte multiple. If set, this parameter is ignored when direct I/O is performed.
Increasing the size of the readahead parameter increases the performance of large file transfers, but only to a point. You should test the performance of the system after resetting the readahead size until you see no more improvement in transfer rates. The following is an example method of testing timings on disk reads:

```
# timex dd if=/sam/myfile of=/dev/null bs=256k
```

You should test various readahead sizes for your environment. The readahead parameter should be set to a size that increases the I/O performance for paged I/O, but is not so large as to hurt performance. It is also important to consider the amount of memory and number of concurrent streams when you set the readahead value. Setting the readahead value multiplied by the number of streams to a value that is greater than memory can cause page thrashing.
The default readahead value is 1024 kilobytes. This value favors large-block, sequential I/O. For short-block, random I/O applications, set readahead to the typical request size. Database applications do their own read-ahead, so for these applications, set readahead to 0.
The readahead setting can be enabled from a mount option, from within the samfs.cmd file, from within the /etc/vfstab file, or from a command within the samu(1M) utility. For information about enabling this setting from a mount option, see the{{– o readahead=}}n option on the mount_samfs(1M) man page. For information about enabling this setting from the samfs.cmd file, see the samfs.cmd(4) man page. For information about enabling this setting from within samu(1M), see the samu(1M) man page.

7. Set the stripe width.
   The{{– o stripe=}}n option with the mount(1M) command specifies the stripe width for the file system. The stripe width is based on the disk allocation unit (DAU) size. The n argument specifies that n x DAU bytes are written to one device before writing switches to the next device. The DAU size is set when the file system is initialized by the sammkfs(1M) -a command.
   If -o stripe=0 is set, files are allocated to file system devices using the round-robin allocation method. With this method, each file is completely allocated on one device until that device is full. Round-robin is the preferred setting for a multistream environment. If{{– o stripe=}}n is set to an integer greater than 0, files are allocated to file system devices using the stripe method. To determine the appropriate -o stripe=n setting, try varying the setting and taking performance readings. Striping is the preferred setting for turnkey applications with a required bandwidth.
   You can also set the stripe width from the /etc/vfstab file or from the samfs.cmd file.
   For more information about the mount(1M) command, see the mount_samfs(1M) man page. For more information about the samfs.cmd file, see the samfs.cmd(4) man page.

Enabling Qwrite Capability

By default, the Sun QFS file systems disable simultaneous reads and writes to the same file. This is the mode defined by the UNIX vnode interface standard, which gives exclusive access to only one write while other writers and readers must wait. Qwrite enables simultaneous reads and writes to the same file from different threads.

The Qwrite feature can be used in database applications to enable multiple simultaneous transactions to the same file. Database applications typically manage large files and issue simultaneous reads and writes to the same file. Unfortunately, each system call to a file acquires and releases a read/write lock inside the kernel. This lock prevents overlapped (or simultaneous) operations to the same file. If the application itself implements file-locking mechanisms, the kernel-locking mechanism impedes performance by unnecessarily serializing I/O.

Qwrite can be enabled in the /etc/vfstab file, in the samfs.cmd file, and as a mount option. The -o qwrite option with the mount(1M) command bypasses the file system locking mechanisms (except for applications accessing the file system through the network file system (NFS)) and lets the application control data access. If qwrite is specified, the file system enables simultaneous reads and writes to the same file from different threads. This option improves I/O performance by queuing multiple requests at the drive level.

The following example uses the mount(1M) command to enable Qwrite on a database file system:

```
# mount -F samfs -o qwrite /db
```

For more information about this feature, see the qwrite directive on the samfs.cmd(4) man page or the -o qwrite option on the mount_samfs(1M) man page.

Setting the Write Throttle

The `-o wr_throttle=n` option limits the number of outstanding write kilobytes for one file to n. By default, Sun QFS file systems set the `wr_throttle` to 16 megabytes.

If a file has n write kilobytes outstanding, the system suspends an application that attempts to write to that file until enough bytes have completed the I/O to allow the application to be resumed.

If your site has thousands of streams, such as thousands of NFS-shared workstations accessing the file system, you can tune the `-o wr_throttle=n` option in order to avoid flushing excessive amounts of memory to disk at once. Generally, the number of streams multiplied by 1024 x the n argument to the `-o wr_throttle=n` option should be less than the total size of the host system's memory minus the memory needs of the Solaris OS, as shown in this formula:
number-of-streams x n x 1024 < total-memory - Solaris-OS-memory-needs

For turnkey applications, you might want to use a size larger than the default 16,384 kilobytes, because this keeps more pages in memory.

Setting the Flush-Behind Rate

Two mount parameters control the flush-behind rate for pages written sequentially and for stage pages. The `flush_behind` and `stage_flush_behind` mount parameters are read from the `samfs.cmd` file, the `/etc/vfstab` file, or the `mount`(1M) command.

The `flush_behind=n` mount parameter sets the maximum flush-behind value. Modified pages that are being written sequentially are written to disk asynchronously to help the Solaris™ Volume Manager (SVM) layer keep pages clean. To enable this feature, set n to be an integer from 16 through 8192. By default, n is set to 0, which disables this feature. The n argument is specified in kilobyte units.

The `stage_flush_behind=n` mount parameter sets the maximum stage flush-behind value. Stage pages that are being staged are written to disk asynchronously to help the SVM layer keep pages clean. To enable this feature, set n to be an integer from 16 through 8192. By default, n is set to 0, which disables this feature. The n argument is specified in kilobyte units.

For more information about these mount parameters, see the `mount_samfs`(1M) man page or the `samfs.cmd`(4) man page.

Tuning the Number of Inodes and the Inode Hash Table

The Sun QFS file system enables you to set the following two tunable parameters in the `/etc/system` file:

- `ninodes`
- `nhino`

To enable non-default settings for these parameters, edit the `/etc/system` file, and then reboot your system.

The following subsections describe these parameters in more detail.

The **ninodes** Parameter

The `ninodes` parameter specifies the maximum number of default inodes. The value for `ninodes` determines the number of in-core inodes that Sun QFS software keeps allocated to itself, even when applications are not using many inodes.

The format for this parameter in the `/etc/system` file is as follows:

```
set samfs:ninodes = _value_
```

The range for value is from 16 through 2000000. The default value for `ninodes` is one of the following:

- A value that is equal to the `ncsize` setting. The `ncsize` parameter is a Solaris tuning parameter that specifies the number of entries in the directory name look-up cache (DNLC). For more information about `ncsize`, see the Solaris Tunable Parameters Reference Manual.
- `2000`. The file systems set `ninodes` to `2000` if the `ncsize` setting is zero or out of range.

The **nhino** Parameter

The `nhino` parameter specifies the size of the in-core inode hash table.

The format for this parameter in the `/etc/system` file is as follows:

```
set samfs:nhino = _value_
```

The range for value is `1` through 1048756. value must be a nonzero power of 2. The default value for `nhino` is one of the following:

- A value that is equal to the ninodes value divided by 8 and then, if necessary, rounded up to the nearest power of 2. For example, assume that the following line exists in `/etc/system`:

```
set samfs:ninodes 8000
```

For this example, if `nhino` is not set, the system assumes `1024`, which is 8000 divided by 8 and then rounded up to the nearest power of 2 (210)

- `512`. The file systems set `nhino` to `512` if the `ninodes` setting is out of range.

When to Set the **`ninodes`** and **`nhino`** Parameters

When searching for an inode by number (after obtaining an inode number from a directory or after extracting an inode number from an NFS file handle), a Sun QFS file system searches its cache of in-core inodes. To speed this process, the file system maintains a hash table to decrease the number of inodes it must check.

A larger hash table reduces the number of comparisons and searches, at a modest cost in memory usage. If the `nhino` value is too large, the system is slower when undertaking operations that sweep through the entire inode list (inode syncs and unmounts). For sites that manipulate large numbers of files and sites that do extensive amounts of NFS I/O, it can be advantageous to set these parameter values to larger than the defaults.

If your site has file systems that contain only a small number of files, it might be advantageous to make these numbers smaller than the defaults. This could be the case, for example, if you have a file system into which you write large single-file `tar`(1) files to back up other file systems.

Using QFS File Systems with SANergy (SAN-QFS)

Using the SAN-QFS File System in a Heterogeneous Computing Environment

The SAN-QFS file system enables multiple hosts to access the data stored in a Sun QFS system at full disk speeds. This capability can be especially useful for database, data streaming, web page services, or any application that demands high-performance, shared-disk access in a heterogeneous environment.

You can use the SAN-QFS file system in conjunction with fibre-attached devices in a storage area network (SAN). The SAN-QFS file system enables high-speed access to data through Sun QFS software and software such as Tivoli SANergy file-sharing software. To use the SAN-QFS file system, you must have both the SANergy (2.2.4 or later) and the Sun QFS software. For information about the levels of Sun QFS and SANergy software that are supported, contact your Sun sales representative.

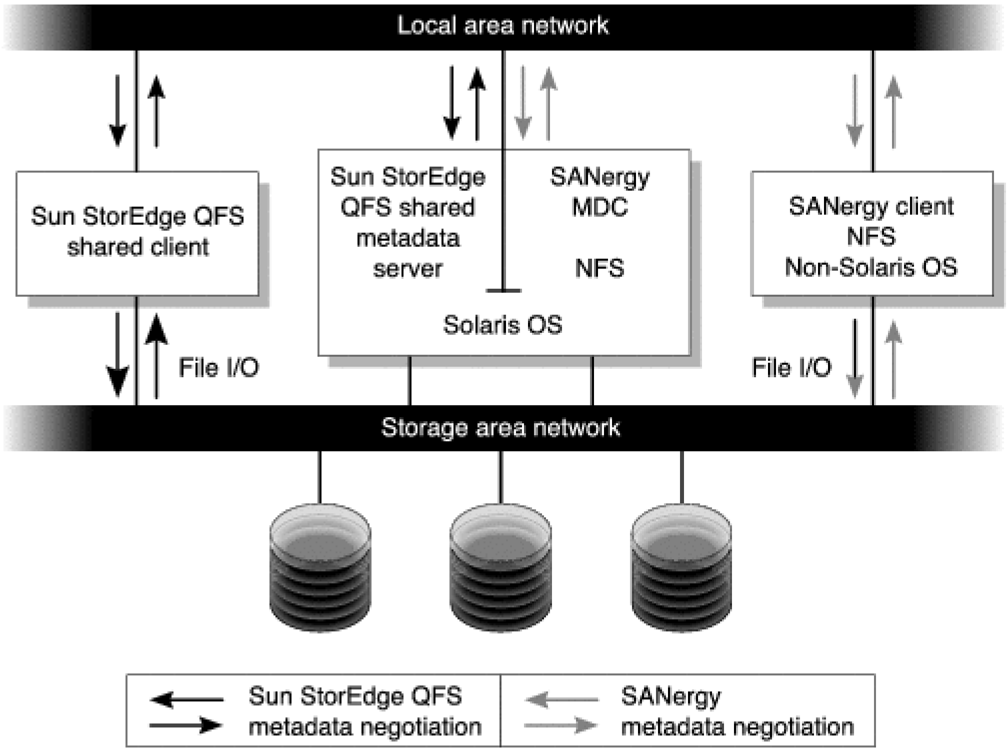
> Note -
> In environments that include the Solaris OS and supported Linux operating systems, use the Sun QFS shared file system, not the SAN-QFS file system, on the Solaris hosts. For information about the Sun QFS shared file system, see Configuring a Shared File System. For a comparison of the Sun QFS shared file system and the SAN-QFS file system, see SAN-QFS Shared File System and Sun QFS Shared File System Comparison.

The following figure depicts a SAN-QFS file system that uses both the Sun QFS software and the SANergy software and shows that the clients and the metadata controller (MDC) system manage metadata across the local area network (LAN). The clients perform I/O directly to and from the storage devices.

Note that all clients running only the Solaris OS are hosting the Sun QFS software, and that all heterogeneous clients running an OS other than Solaris are hosting the SANergy software and the NFS software. The SAN-QFS file system's metadata server hosts both the Sun QFS and the SANergy software. This server acts not only as the metadata server for the file system but also as the SANergy MDC.

> **Note -**
> The SANergy software is not supported on x64 hardware platforms.

Figure – SAN-QFS File System Using Sun QFS Software and SANergy Software



Enabling the SAN-QFS File System

The following procedures describe how to enable the SAN-QFS file system. Perform these procedures in the order in which they are presented.

Before You Begin

Before you enable the SAN-QFS file system, keep the following configuration considerations in mind and plan accordingly:

- Disks configured for use in a SAN-QFS file system cannot be under the control of a volume manager.
- For the Sun QFS metadata server to be enabled or relocated in a SAN-QFS environment, the new metadata server system must be configured as a SANergy metadata controller.
- A SAN-QFS file system does not recognize segmented files. This can result in unexpected behavior if segmented files are used within the SAN-QFS environment.
- Devices that are classified as `ms` or `md` devices in the Sun QFS `mcf` file are not supported in a SAN-QFS file system.

> **Note -**
> This documentation assumes that your non-Solaris clients are hosting SANergy software and NFS software for file system sharing. The text and examples in this document reflect this configuration. If your non-Solaris clients host the Samba software instead of the NFS software, see your Samba documentation.

How to Enable the SAN-QFS File System on the Metadata Controller

When you use the SAN-QFS file system, one host system in your environment acts as the SANergy metadata controller (MDC). This is the host system upon which the Sun QFS file system resides.

1. Log in to the host upon which the Sun QFS file system resides and become `superuser`.

2. Verify that the Sun QFS file system is tested and fully operational.
3. Install and configure the SANergy software.
   For instructions, see your SANergy documentation.
4. Use the pkginfo(1) command to verify the SANergy software release level:

```
# pkginfo -l SANergy
```

5. Ensure that the file system is mounted.
   Use the mount(1M) command either to verify the mount or to mount the file system.
6. Use the share(1M) command in the following format to enable NFS access to client hosts:

```
MDC# share -F nfs -d _qfs-file-system-name_ _/mount-point_
```

   For qfs-file-system-name, specify the name of your Sun QFS file system, such as qfs1. For more information about the share(1M) command, see the share(1M) or share_nfs(1M) man page.
   For mount-point, specify the mount point of qfs-file-system-name.

7. If you are connecting to Microsoft Windows clients, configure Samba, rather than NFS, to provide security and namespace features. To do this, add the SANERGY_SMBPATH environment variable in the /etc/init.d/sanergy file and point it to the location of the Samba configuration file. For example, if your Samba configuration file is named /etc/swf/smb.conf, you must add the following lines to the beginning of your /etc/init.d/sanergy file:
   SANERGY_SMBPATH=/etc/sfw/smb.confexport SANERGY_SMBPATH
8. (Optional) Edit the file system table (/etc/dfs/dfstab) on the MDC to enable access at boot time.
   Perform this step if you want to automatically enable this access at boot time.

How to Enable the SAN-QFS File System on the Clients

After you have enabled the file system on the MDC, you are ready to enable it on the client hosts. The SAN-QFS file system supports several client hosts including IRIX, Microsoft Windows, AIX, and Linux hosts. For information about the specific clients supported, see your Sun sales representative.

Every client has different operational characteristics. This procedure uses general terms to describe the actions you must take to enable the SAN-QFS file system on the clients. For information specific to your clients, see the documentation provided with your client hosts.

Steps

1. Log in to each of the client hosts.
2. Edit the file system defaults table on each client and add the file system.
   For example, on a Solaris OS, edit the /etc/vfstab file on each client and add the name of your Sun QFS file system, as follows:

```
server:/qfs1  -  /qfs1  nfs  -  yes  noac,hard,intr,timeo=1000
```

   On other operating system platforms, the file system defaults table might reside in a file other than /etc/vfstab. For example, on Linux systems, this file is /etc/fstab.
   For more information about editing the /etc/vfstab file, see Sun SAM-QFS Installation and Upgrade Guide. For information about required or suggested NFS client mount options, see your SANergy documentation.

How to Install the SANergy Software on the Clients

After enabling the file system on the client hosts, you are ready to install the SANergy software on the clients. The following procedure describes the SANergy installation process in general terms.

1. Install and configure the SANergy software.
   For instructions, see your SANergy documentation.
2. Use the mount command to NFS mount the file system.
   For example:

```
# mount <host>:/<mount-point>/ <local-mount-point>
```

For host, specify the MDC.
For mount-point, specify the mount point of the Sun QFS file system on the MDC.
For local-mount-point, specify the mount point on the SANergy client.

3. Use the SANergy `fuse` command to fuse the software:

```
#  fuse |<mount-point>
```

For mount-point, specify the mount point on the SANergy client.

Unmounting the SAN-QFS File System

The following procedures describe how to unmount a SAN-QFS file system that is using the SANergy software. Perform these procedures in the order in which they are presented.

How to Unmount the SAN-QFS File System on the SANergy Clients

Follow these steps for each client host on which you want to unmount the SAN-QFS file system.

1. Log in to the client host and become `superuser`.
2. Use the SANergy `unfuse` command to unfuse the file system from the software:

```
# unfuse|<mount-point>
```

For mount-point, specify the mount point on the SANergy client.

3. Use the `umount`(1M) command to unmount the file system from NFS:

```
# umount <host>:/<mount-point>/ <local-mount-point>
```

For host, specify the MDC.
For mount-point, specify the mount point of the Sun QFS file system on the MDC.
For local-mount-point, specify the mount point on the SANergy client.

How to Unmount the SAN-QFS File System on the Metadata Controller

1. Log in to the MDC system and become `superuser`.
2. Use the `unshare`(1M) command to disable NFS access to client hosts:

```
MDC# unshare <qfs-file-system-name> <mount-point>
```

For qfs-file-system-name, specify the name of your Sun QFS file system, such as `qfs1`. For more information about the `unshare(1M)` command, see the share(1M) man page.
For mount-point, specify the mount point of qfs-file-system-name.

How to Unmount the SAN-QFS File System on the Sun QFS Clients

Follow these steps on each participating client host.

1. Log in to a Sun QFS client host and become `superuser`.
2. Use the `umount`(1M) command to unmount the file system.
   For example:

```
# umount /qfs1
```

How to Unmount the SAN-QFS File System on the Sun QFS Server

1. Log in to the host system upon which the Sun QFS file system resides and become `superuser`.
2. Use the `umount`(1M) command to unmount the file system.

Troubleshooting: Unmounting a SAN-QFS File System With SANergy File Holds

SANergy software issues holds on Sun QFS files to reserve them temporarily for accelerated access. If SANergy crashes when holds are in effect, you will not be able to unmount the file system. If you are unable to unmount a SAN-QFS file system, examine the `/var/adm/messages` file and look for console messages that describe outstanding SANergy holds.

Whenever possible, allow the SANergy file-sharing function to clean up its holds, but in an emergency, or in case of a SANergy file-sharing system failure, use the following procedure to avoid a reboot.

How to Unmount a File System in the Presence of SANergy File Holds

1. Use the `unshare`(1M) command to disable NFS access.
2. Use the `samunhold`(1M) command to release the SANergy file system holds.
   For more information about this command, see the `samunhold`(1M) man page.
3. Use the `umount`(1M) command to unmount the file system.

Block Quotas in a SAN-QFS File System

The SANergy software does not enforce block quotas. Therefore, it is possible for you to exceed a block quota when writing a file with the SANergy software. For more information on quotas, see Enabling Quotas.

File Data and File Attributes in a SAN-QFS File System

The SANergy software uses the NFS software for metadata operations, which means that the NFS close-to-open consistency model is used for file data and attributes. File data and attributes among SANergy clients do not support the POSIX coherency model for open files.

Using **`samgrowfs`**(1M) to Expand SAN-QFS File Systems

You can use the `samgrowfs`(1M) command to increase the size of a SAN-QFS file system. To perform this task, follow the procedures described in Adding Disk Cache to a File System.

> Caution -
> When using this procedure, be aware that the line-by-line device order in the `mcf` file must match the order of the devices listed in the file system's superblock.

When the `samgrowfs`(1M) command is issued, the devices that were already in the `mcf` file keep their positions in the superblock. New devices are written to subsequent entries in the order in which the are encountered.

If this new order does not match the order in the superblock, the SAN-QFS file system cannot be fused.

SAN-QFS Shared File System and Sun QFS Shared File System Comparison

The SAN-QFS shared file system and the Sun QFS shared file system have the following similarities:

- Both can stage files.
- Both are useful in data capture environments in which it is desirable that the primary file system host not be responsible for writing the data.
- Both are advantageous in environments where there is contention for writing files.

The following table describes differences between the file systems.

Table – SAN-QFS Shared File System Versus Sun QFS Shared File System

SAN-QFS File System	Sun QFS Shared File System
Uses NFS protocol for metadata.	Uses natural metadata.
Preferred in heterogeneous computing environments (that is, when not all hosts are Sun systems).	Preferred in homogeneous Solaris OS environments.
Useful in environments where multiple, heterogeneous hosts must be able to write data.	Preferred when multiple hosts must write to the same file at the same time.

Mount Options in a Shared File System

The Sun QFS shared file system can be mounted with several mount options. This section describes some of these options within the context of their roles.

You can specify most mount options by using the `mount`(1M) command, by entering them in the `/etc/vfstab` file, or by entering them in the `samfs.cmd` file. For example, the following `/etc/vfstab` file includes `mount`(1M) options for a shared file system:

```
sharefs1 - /sfs samfs - no shared,mh_write
```

You can change some mount options dynamically by using the `samu`(1M) operator utility. For more information about these options, see Using the samu Operator Utility.

The following sections summarize the mount options available to you in a shared file system. For more information about any of these mount options, see the `mount_samfs`(1M) man page or see the cross-references mentioned in their descriptions.

Mounting in the Background: the **bg** Option

The bg mount option specifies that if the first mount operation fails, subsequent attempts at mounting should occur in the background. By default, bg is not in effect, and mount attempts continue in the foreground.

Reattempting a File System Mount: the **`retry`** Option

The `retry` mount option specifies the number of times that the system should attempt to mount a file system. The default is 10000.

Declaring a Sun QFS Shared File System: the **`shared`** Option

The `shared` mount option declares a file system to be a Sun QFS shared file system. This option must be specified in the `/etc/vfstab` file in order for the file system to be mounted as a Sun QFS shared file system. The presence of this option in a `samfs.cmd` file or on the `mount`(1M) command does not cause an error condition, but it does not mount the file system as a shared file system.

Tuning Allocation Sizes: the **minallocsz=**n and **maxallocsz=**n Options

The `-o minallocsz=n` and `-o maxallocsz=n` options to the `mount`(1M) command specify an amount of space, in kilobytes. This is the minimum block allocation size. If a file is growing, the metadata server allocates blocks when an append lease is granted. You can use the `-o minallocsz=n` option to specify the initial size of this allocation. The metadata server can increase the size of the block allocation depending on the application's access patterns up to, but not exceeding, the `-o maxallocsz=n` option's setting.

You can specify these `mount`(1M) options on the `mount`(1M) command line, in the `/etc/vfstab` file, or in the `samfs.cmd` file.

Using Leases in a Sun QFS Shared File System: the **rdlease=**n, **wrlease=**n, and **aplease=**n Options

A lease grants a shared host permission to perform an operation on a file for as long as the lease is valid. The metadata server issues leases to each shared host, including itself. The leases are renewed as necessary to permit continued file operations. The possible file operations are as follows:

- A read lease enables existing file data to be read.
- A write lease enables existing file data to be overwritten.
- An append lease enables a file's size to be extended and enables newly allocated blocks to be written.

A shared host can continue to update leases for as long as necessary. The lease is transparent to the end user. The following table shows the mount options that enable you to specify the duration of each lease type.

Table – Lease-Related ***mount***(1M) Options

Option	Action
`-o rdlease=n`	This option specifies the maximum amount of time, in seconds, for the read lease.
`-o wrlease=n`	This option specifies the maximum amount of time, in seconds, for the write lease.
`-o aplease=n`	This option specifies the maximum amount of time, in seconds, for the append lease.

All three leases enable you to specify an n such that $15 \leq n \leq 600$. The default time for each lease is 30 seconds. A file cannot be truncated if a lease is in effect. For more information about setting these leases, see the `mount_samfs`(1M) man page.

If you change the metadata server because the current metadata server is down, you must add the lease time to the changeover time because all leases must expire before an alternate metadata server can assume control.

Setting a short lease time causes more traffic between the client hosts and the metadata server because the lease must be renewed after it has expired.

Enabling Multiple Host Reads and Writes: the **mh_write** Option

By default, in a Sun QFS shared file system, multiple hosts can read the same file at the same time, and if no host is writing to that file, I/O can be paged on all hosts. Only one host can append or write to a file at any one time.

The `mh_write` option controls write access to the same file from multiple hosts. If `mh_write` is specified as a mount option on the metadata server host, the Sun QFS shared file system enables simultaneous reads and writes to the same file from multiple hosts. If `mh_write` is not specified on the metadata server host, only one host can write to a file at any one time.

By default, `mh_write` is disabled, and only one host has write access to a file at any one time. The length of that time period is determined by the duration of the `wrlease` mount option. If the Sun QFS shared file system is mounted on the metadata server with the `mh_write` option enabled, simultaneous reads and writes to the same file can occur from multiple hosts.

The following table describes how file access from multiple hosts is affected depending on whether the `mh_write` option is enabled on the metadata server.

Table – File Access Based on the ***mh_write*** Option

mh_write Not Enabled on the Metadata Server	**mh_write** Enabled on the Metadata Server

Multiple reader hosts allowed.Can use paged I/O.	Multiple reader hosts allowed.Can use paged I/O.
Only one writer host is allowed.Can use paged I/O.All other hosts wait.	Multiple reader and/or writer hosts allowed.If any writer hosts exist, all I/O is direct.
Only one append host.All other hosts wait.	Only one append host is allowed.All other hosts can read and/or write.If any writer hosts exist, all I/O is direct.

The `mh_write` option does not change locking behavior. File locks behave the same whether `mh_write` is in effect or not. The `mh_write` option's effect is as follows:

- When `mh_write` is in effect, all hosts can read from and write to the same file simultaneously.
- When `mh_write` is not in effect, only one host can write to a given file during a given time interval, and no hosts can read from the file during that time interval.

Sun QFS shared file system maintains consistency between hosts. The first time that a host executes a read or write system call, it gets a lease, which enables it to read or write the file for some period of time. The existence of that lease prevents other hosts without `mh_write` from accessing the file. In particular, the lease can last longer than the duration of the system call that caused its acquisition.

When `mh_write` is not in effect, the Sun QFS shared file system should provide near-POSIX behavior for data reads and writes. For metadata, however, access time changes might not be seen immediately on other hosts. Changes to a file are pushed to disk at the end of a write lease, and when a read lease is acquired, the system invalidates any stale cache pages so that the newly written data can be seen.

When `mh_write` is in effect, behavior might be less consistent. When there are simultaneous readers and writers, the Sun QFS shared file system switches all hosts accessing the file into direct I/O mode. This means that page-aligned I/O should be visible immediately to other hosts. However, non-page-aligned I/O can result in stale data being visible, or even written to the file, because the normal lease mechanism that prevents this has been disabled.

You should specify the `mh_write` option only when multiple hosts need to write to the same file simultaneously and when applications perform page-aligned I/O. In other cases, there is some risk of data inconsistency because even using `flock()` (which works with `mh_write`) to coordinate between hosts does not guarantee consistency.

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

Setting the Minimum Number of Concurrent Threads: the **`min_pool=`**n Option

The `min_pool=`n mount option sets the minimum number of concurrent threads for the Sun QFS shared file system. By default, `min_pool=64` on Solaris systems. This means that using default settings, there will always be at least 64 active threads in the thread pool on Solaris and 8 on Linux. You can adjust the `min_pool=`n mount option to any value between 8 and 2048, depending on the Sun QFS shared file system's activity.

The `min_pool` mount option must be set in the `samfs.cmd` file. It will be ignored if set in the `/etc/vfstab` file or on the command line.

> Note -
> The `min_pool` mount option replaces the previous `nstreams` mount option. In version 5.0 of the software, the `nstreams` option is completely removed.

Retaining Cached Attributes: the **`meta_timeo=`**n Option

The `meta_timeo=`n mount option determines how long the system waits between checks on the metadata information. By default, the system refreshes metadata information every three seconds. This means, for example, that an `ls`(1) command entered in a Sun QFS shared file system with several newly created files might not return information about all the files until three seconds have passed. For n, specify a value such that $0 \leq n \leq 60$.

Specifying Striped Allocation: the **`stripe`** Option

By default, data files in the Sun QFS shared file system are allocated using the round-robin file allocation method. To specify that file data be striped across disks, you can specify the `stripe` mount option on the metadata host and all potential metadata hosts. Note that by default, unshared file systems allocate file data using the striped method.

In a round-robin allocation, files are created in a round-robin fashion on each slice or striped group. This causes the maximum performance for one file to be the speed of a slice or striped group. For more information about file allocation methods, see File System Design Basics.

Specifying the Frequency With Which Metadata Is Written: the **sync_meta=**n Option

You can set the sync_meta=n option to sync_meta=1 or sync_meta=0.

By default, sync_meta=1 and a Sun QFS shared file system writes file metadata to disk every time the metadata changes. This slows data performance, but it ensures data consistency. This is the setting that must be in effect if you want to change the metadata server.

If you set sync_meta=0, the Sun QFS shared file system writes the metadata to a buffer before writing it to disk. This delayed write delivers higher performance, but it decreases data consistency after an unscheduled machine interruption.

Enabling WORM Functionality: the **worm_capable** and **def_retention** Options

If you are using the optional WORM package, the worm_capable mount option enables the file system to support WORM files. The def_retention mount option sets the default retention time using the format {{def_retention=}}MyNdOhPm.

In this format, M, N, O, and P are non-negative integers and y, d, h, and m stand for years, days, hours, and minutes, respectively. Any combination of these units can be used. For example, 1y5d4h3m indicates 1 year, 5 days, 4 hours, and 3 minutes; 30d8h indicates 30 days and 8 hours; and 300m indicates 300 minutes. This format is backward compatible with the formula in previous software versions, in which the retention period was specified in minutes.

See Configuring WORM-FS File Systems for more information about the WORM functionality.

Using the **samu**(1M) Operator Utility

This section shows how to use samu(1M) to control the devices configured within your environment. Many samu(1M) displays are useful only for sites using the storage and archive management mechanism. If you are using samu(1M) in a Sun QFS-only environment, these displays do not apply to you.

You can also use the samcmd(1M) command to perform many of the same operations. For more information, see the samcmd(1M) man page.

Operator Utility At a Glance

Option	Description	For More Information
a	Displays archiver status	(a) - Archiver Status Display
c	Displays device configuration	(c) - Device Configuration Display
C	Displays memory information	(C) - Memory Display
d	Displays traced events	(d) - Daemon Trace Controls Display
D	Displays the disk volume dictionary	(D) - Disk Volume Dictionary
f	Displays the components of a Sun QFS file systems	(f) - File Systems Display
F	Displays the label on an optical disk	(F) - Optical Disk Label Display
h	Displays a summary of the available samu(1M) displays	(h) - Help Display
I	Displays the content of inodes	(I) - Inode Display
J	Displays the shared memory segment for the preview queue	(J) - Preview Shared Memory Display
K	Displays kernel statistics	(K) - Kernel Statistics Display
l	Displays usage information for the file system	(l) - Usage Display
L	Displays the location of shared memory tables and system defaults in shared memory	(L) - Shared Memory Tables

m	Displays the status of mass storage mounted file systems and member drives	(m) - Mass Storage Status Display
N	Displays information about the file system	(N) - File System Parameters Display
o	Displays the status of optical disk drives	(o) - Optical Disk Status Display
p	Displays pending load requests for removable media	(p) - Removable Media Load Requests Display
P	Displays the services registered with the Sun QFS single port multiplexer	(P) - Active Services Display
r	Displays the activity of removable media devices	(r) - Removable Media Status Display
R	Displays information and status of Sun SAM-Remote configurations	(R) - Sun SAM-Remote Information Display
s	Displays the status for configured devices	(s) - Device Status Display
S	Displays raw device data	(S) - Sector Data Display
t	Displays the satus of all configured tape drives	(t) - Tape Drive Status Display
T	Displays the SCSI status of a SCSI device	(T) - SCSI Sense Data Display
u	Displays the list of files in the staging queue	(u) - Staging Queue Display
U	Displays the device table in a human-readable form	(U) - Device Table Display
v	Displays the location and VSN of all cataloged disks or tapes	(v) - Automated Library Catalog Display
w	Displays pending stage requests with unloaded volumes	(w) - Pending Stage Queue
down	Terminates operation on device	Device Commands
idle	Restricts access to device by preventing new connections	Device Commands
off	Logically turns off device	Device Commands
on	Logically turns on device	Device Commands
unavail	Makes device unavailable for use with the file system	Device Commands
unload	Unloads the mounted media for the specified removable media device	Device Commands
nalloc	Prohibits any future allocation to the device	Device Commands
alloc	Re-enables allocation to the device	Device Commands
flush_behind	Controls whether modified pages are written to disk asynchronously	The flush_behind Command
force_nfs_async noforce_nfs_async	Controls whether the file system caches NFS data written to the server	The force_nfs_async and noforce_nfs_async Commands
sw_raid nosw_raid	Controls whether the file system aligns the writebehind buffer.	The sw_raid and nosw_raid Commands
readahead	Sets the maximum number of bytes that can be read ahead by the file system	The readahead Command
writebehind	Sets the maximum number of bytes that can be written behind by a file system.	The writebehind Command
wr_throttle	Sets the number of outstanding write kilobytes for one file.	The wr_throttle Command
dio_rd_form_min	Sets the minimum number of blocks for read operations for well-aligned I/O	The dio_rd_form_min and dio_wr_form_min Commands
dio_wr_form_min	Sets the minimum number of blocks for write operations for well-aligned I/O	The dio_rd_form_min and dio_wr_form_min Commands
dio_rd_ill_min	Sets the minimum number of blocks for read operations for misaligned I/O	The dio_rd_ill_min and dio_wr_ill_min Commands

dio_wr_ill_min	Sets the minimum number of blocks for write operations for misaligned I/O	The dio_rd_ill_min and dio_wr_ill_min Commands
dio_rd_consec	Sets the number of consecutive read operations that can occur without regard to a buffer size	The dio_rd_consec and dio_wr_consec Commands
dio_wr_consec	Sets the number of consecutive write operations that can occur without regard to a buffer size	The dio_rd_consec and dio_wr_consec Commands
dio_szero nodio_szero	Controls whether uninitialized areas of sparse files written with direct I/O are zeroed when the area is accessed.	The dio_szero and nodio_szero Commands
forcedirectio noforcedirectio	Controls whether direct I/O or buffered I/O is used as the default mode.	The forcedirectio and noforcedirectio Commands
meta_timeo	Sets the time limit for the Sun QFS shared file system metadata cache	The meta_timeo interval Command
mh_write nomh_write	Controls whether to allow multihost read and write operations	The mh_write and nomh_write Commands
minallocsz	Sets the minimum number of allocated blocks	The minallocsz value and maxallocsz value Commands
maxallocsz	Sets the maximum number of allocated blocks	The minallocsz value and maxallocsz value Commands
rdlease	Sets the amount of time for read leases	The rdlease interval, wrlease interval, and aplease interval Commands
wrlease	Sets the amount of time for write leases	The rdlease interval, wrlease interval, and aplease interval Commands
aplease	Sets the amount of time for append leases	The rdlease interval, wrlease interval, and aplease interval Commands
abr noabr	Controls the application binary recovery (ABR) mount option.	The abr and noabr Commands
dmr nodmr	Controls the direct mirror reads (DMR) mount option.	The dmr nd nodmr Commands
invalid	Sets the number of seconds that the file system holds cached attributes	The invalid interval Command
mm_stripe	Sets the number for disk allocation units for the metadata stripe width	The mm_stripe Command
qwrite noqwrite	Controls whether read and write operations to the same file are performed simultaneously from different threads.	The qwrite and noqwrite Commands
refresh_at_eof norefresh_at_eof	Controls whether Sun QFS hosts update file sizes	The refresh_at_eof and norefresh_at_eof Commands
suid nosuid	Controls whether running programs can change their owner IDs.	The suid and nosuid Commands
stripe	Sets the stripe width for the file system to the specified number of disk allocation units	The stripe Command
sync_meta	Controls whether metadata is written to disk immediately	The sync_meta value Command
trace notrace	Controls whether a file system uses the trace feature	The trace and notrace Commands
clear	Removes the specified volume from the Removable Media Mount Requests display	The clear vsn Command
devlog	Specifies one or more events to be logged	The devlog Command
diskvols	Controls the flags in the disk volume dictionary	The diskvols Command
dtrace	Control the dtrace feature for one or more processes	The dtrace Command

fs	Sets the default file system	The fs Command
mount	Specifies a Sun QFS file system	The mount Command
open	Enables access to a disk device	The open Command
read	Reads the specified sector from an open disk device	The read Command
refresh	Sets the amount of time between samu(1M) screen refreshes	The refresh Command
snap	Sends a copy of an operator display to file	The snap Command
!	Enables you to run a shell command within the samu(1M) operator utility.	The ! shell_command Command

Overview

The samu(1M) operator utility requires a display terminal that displays a minimum of 24 lines by 80 characters wide. The utility includes the following features:

- Displays that enable you to monitor Sun QFS and SAM-QFS devices and file system activity.
- Commands that enable you to select displays, set display options, control access to and the activity of devices, and take snapshots of display windows.
- Commands that enable you to tune a running file system.

The display windows shown in this section are representative examples. The format and amount of information displayed on your terminal can be different, depending on your terminal model and the devices configured in your environment.

The following sections describe how to start and stop samu(1M), interact with the utility, access the help windows, and view operator displays.

How to Invoke **samu**(1M)

To start the operator utility, type the samu(1M) command from the command line:

```
# samu
```

The system starts samu(1M) and shows the help display. This is the default initial display. To view a different samu(1M) display, follow the steps in How to Display a samu(1M) Screen.
To change the default initial display, see the samu(1M) man page.

> Note -
> samu(1M), like the vi(1) editor, is based on the curses(3CURSES) routine. If you have trouble invoking samu(1M), make sure that your terminal type is defined correctly.

How to Display a **samu**(1M) Screen

The samu(1M) command accepts options on its command line for displaying different samu(1M) screens.

1. Type a colon (:) character.
   The following appears in the lower left corner:

   ```
   Command:
   ```

2. Type the letter that corresponds to the display you want to view and press the Return key.
   For example, to view the Automated Library Catalog Display, type a v character and press Return.
   For a complete list of the displays and the letters that correspond to the displays, enter (h) - Help Display.

How to Stop **samu**(1M)

To exit samu(1M), type one of the following:

- q
- :q
  The samu(1M) operator utility exits and returns you to the command shell.

Interacting With **samu**(1M)

The samu(1M) utility is similar the UNIX vi(1) editor with respect to paging forward or backward, entering commands, refreshing the display, and quitting the utility.

For each operator display, the description includes the control key sequences you use to navigate in that display. The samu(1M) man page summarizes the control key sequences.

The last line of the display window shows the error messages. If a command error occurs, automatic display refreshing halts until the next operator action.

Entering a Device

Each device included in the Sun QFS environment is assigned an Equipment Number (for example, 10) in the mcf file. Many samu(1M) commands require you to identify a specific device using its Equipment Number. To see a list of the devices and their Equipment Numbers, see the (c) - Device Configuration Display

Example
The syntax for the :off command is as follows:

```
:off <eq>
```

For eq, type the Equipment Number for the device you are trying to address.

How to Access Online Help

When you start samu(1M), the default display is the first page of the online Help. For more information about the help (h) display, see (h) - Help Display.

1. Type :h
   To move forward or backward from one screen to the next, type the following key sequence:
   - Press Ctrl-f to move to the following page
   - Press Ctrl-b to move backward to previous page.
2. To return to the help display at any time, pressing the h key.

Operator Displays

To view one of the operator displays, press the its corresponding character key. The lowercase characters, a through w, display operational information.

> Note -
> The uppercase samu(1M) displays (C, D, F, I, J, K, L, M, N, P, R, S, T, and U) are designed to be used at a customer site only with the assistance of a member of the technical support staff.
> This topic does not describe these uppercase displays in detail.

When a operator display overflows the available screen area, the word more appears on the bottom of the display indicating additional information, as shown in the following example|#gftiu]. Use Ctrl-f key sequence to page forward and see more content.

Example – ***samu***(1M) Screen Indicating More Text Can Be Obtained

```
xb54  54  exb8505  pt03  0  yes  2  0  on
lt55  55  dlt2000  pt02  1  yes  4  0  on  ml65
hp56  56  hpc1716  pt01  1  yes  3  0  on  hp70
hp57  57  hpc1716  pt01  1  yes  4  0  on  hp70
more
```

When samu(1M) prompts you to enter a device, enter its associated Equipment Number. The configuration display (c) shows Equipment Numbers for all removable media devices. To control all displays, use the control keys listed for the display.

The following sections describe the operator displays in alphabetical order. Examples are provided, and when necessary, displays are followed by a table describing the fields displayed.

> Note -
> If you are using only the Sun QFS software, without the archiving features of SAM-QFS, only some of the `operator` displays will be available.

(**a**) - Archiver Status Display

The a display shows the archiver status.

- To show the status of the archiver for each file-system, type the command with the following format:

```
Command:a
```

- To display archiving details for a specific file system, type the command with the following format:

```
Command:a <filesystem>
```

For filesystem, specify the name of a file system.

Navigation

The following table shows the control keys you can use in the a display.

Control Keys for the *a* Display

Key	Function
Ctrl-b	Previous file system
Ctrl-f	Next file system
Ctrl-d	Page arcopies forward (bottom portion)
Ctrl-u	Page arcopies backward (bottom portion)

The following table shows the control keys you can use in the `:a <filesystem>` display.

Control Keys for the ***:a <filesystem>*** Display

Key	Function
Ctrl-b	Previous file system
Ctrl-f	Next file system

Sample Display

This example shows activity and statistics for a single file system in the summary display.

Example: ***samu***(1M) ***a*** Display

```
Archiver Status
samu 5.0 07:44:02 August 8 2008
sam-archiverd:  Waiting for resources
sam-arfind:  samfs1 mounted at /sam1
Waiting until 2005-05-08 07:54:02 to scan .inodes
sam-arfind:  samfs2 mounted at /sam2
Waiting until 2005-05-08 07:52:57 to scan .inodes
sam-arfind:  qfs1 mounted at /qfs1
Waiting until 2005-05-08 07:44:33 to scan .inodes
sam-arfind:  qfs2 mounted at /qfs2
Waiting until 2005-05-08 07:53:21 to scan .inodes
sam-arfind:  qfs3 mounted at /qfs3
Waiting until 2005-05-08 07:44:11 to scan .inodes

sam-arfind:  qfs4 mounted at /qfs4
Waiting until 2005-05-08 07:53:35 to scan .inodes

sam-arfind:  shareqfs1 mounted at /shareqfs1
Shared file system client.  Cannot archive.

sam-arfind:  shareqfs2 mounted at /shareqfs2
Shared file system client.  Cannot archive.

sam-arcopy:  qfs4.arset5.1.83 dt.DAT001
Waiting for volume dt.DAT001
```

Field Descriptions

The following table shows the fields in the detail display.

samu a Display Field Descriptions

Field	Description
`samfs1 mounted at`	Mount point.
`regular files`	Number of regular files and their total size.
`offline files`	Number of offline files and their total size.
`archdone files`	Number of `archdone` files and size. Indicates that the archiver has completed processing and can perform no further processing. Files marked as `archdone` have been processed for archiving but have not necessarily been archived.
`copy1`	Number of files and total size for archive copy 1.
`copy2`	Number of files and total size for archive copy 2.
`copy3`	Number of files and total size for archive copy 3.
`copy4`	Number of files and total size for archive copy 4.
`Directories`	Number of directories and total size.
`sleeping until`	Indicates when archiver runs again.

(**c**) - Device Configuration Display

The c display shows your configuration's connectivity. It lists all device names and Equipment Numbers. To invoke the device configuration display, type the command with the following format:

```
Command:c
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the *c* Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-u	Half-page backward

Sample Display

This example shows the device configuration display.

Example:***samu c*** Display

```
Device configuration:                  samu     5.0 07:48:11 Sept 8 2008
ty  eq state   device_name                        fs family_set
sk  100 on      /etc/opt/SUNWsamfs/dcstkconf      100 dcL700
tp  120 off     /dev/rmt/1cbn                     100 dcL700
sg  130 on      /dev/rmt/4cbn                     100 dcL700
sg  140 on      /dev/rmt/5cbn                     100 dcL700
tp  150 off     /dev/rmt/3cbn                     100 dcL700
hy  151 on      historian                         151
```

Field Descriptions

The following table shows the field descriptions for this display.

samu c Display Field Descriptions

Field	Description
`ty`	Device type.
`eq`	Equipment Number of the device.
`state`	Current operating state of the device. Valid device states are as follows: • on The device is available for access. • `ro` The device is available for read-only access. • `off` The device is not available for access. • down The device is available only for maintenance access. • `idle` The device is not available for new connections. Operations in progress continue until completion. • `nalloc` The `nalloc` flag has been set, which prohibits any future allocation to this device. For more information, see Per-logical unit number (LUN) Allocation Control.
`device_name`	Path to the device.

fs	Family Set Equipment Number.
family_set	Name of the storage Family Set or library to which the device belongs.

(C) - Memory Display

The C display shows the content of a specific memory address. To show the content at an address, enter the address in hexadecimal.
To invoke this display, type the following command:

```
Command:C <hex-address>
```

For hex-address, specify the address of a memory location in hexadecimal. For example:

```
Command:C 0x1044a998
```

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This example shows the memory display. The sample output has been truncated.

Example:***samu C*** Display

```
Memory    base: 0x1234567               samu 5.0 07:52:25 Sept 8 2008
00000000  80921000 137ff801 edd05e21 40853420  ......x.mP^!@.4
00000010  00a00420 018528b0 01a604e0 02840640  . . ..(0.&amp;.&rdquo;...@
00000020  02d030a1 a0853420 0080a0a0 100a6fff  .P0! .4 ..  ..o.
00000030  f6921000 13c65e23 582d0000 0ba01020  v....F^#X-... .
00000040  00c45e20 48c608e0 2fd05e21 40920080  .D^ HF.&rdquo;/P^!@...
00000050  037ff801 fa941000 16c45e20 48a600a0  ..x.z....D^ H&amp;.
00000060  80921000 137ff801 d5d05e21 40853420  ......x.UP^!@.4
00000070  00a00420 018528b0 01a604e0 02840640  . . ..(0.&amp;.&rdquo;...@
00000080  02d030a1 c0853420 0080a0a0 100a6fff  .P0!@.4 ..  ..o.
00000090  f6921000 13c65e23 58a01020 00c45e20  v....F^#X . .D^
000000a0  48c608e0 2fd05e21 40920080 037ff801  HF.&rdquo;/P^!@.....x.
000000b0  e39405a2 00c45e20 48a600a0 80921000  c..".D^ H&amp;. ....
000000c0  137ff801 bed05e21 40853420 00a00420  ..x.>P^!@.4 . .
000000d0  018528b0 01a604e0 02840640 02d030a1  ..(0.&amp;.&rdquo;...@.P0!
000000e0  e0853420 0080a0a0 100a6fff f6921000  &rdquo;.4 ..  ..o.v...
000000f0  13c65e23 58a01020 00c45e20 48c608e0  .F^#X . .D^ HF.&rdquo;
```

(d) - Daemon Trace Controls Display

The d display shows the events being traced as specified in the `defaults.conf` file. For more information about enabling trace files, see the `defaults.conf`(4) man page.
To invoke this display, type the following command:

```
Command:d
```

Sample Display

This example shows trace file information. It includes information about the daemons being traced, the paths to the trace files, the events being traced, and information about the size and age of the trace files.

Example:***samu d*** Display

```
Daemon trace controls                   samu 5.0 07:56:38 Sept 8 2008
sam-amld       /var/opt/SUNWsamfs/trace/sam-amld
               cust err fatal misc proc debug date
               size    0    age 0
sam-archiverd  /var/opt/SUNWsamfs/trace/sam-archiverd
               cust err fatal misc proc debug date
               size    0    age 0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
               cust err fatal misc proc debug date
               size    0    age 0
sam-fsd        /var/opt/SUNWsamfs/trace/sam-fsd
               cust err fatal misc proc debug date
               size    0    age 0
sam-rftd       /var/opt/SUNWsamfs/trace/sam-rftd
               cust err fatal misc proc debug date
               size    0    age 0
sam-recycler   /var/opt/SUNWsamfs/trace/sam-recycler
               cust err fatal misc proc debug date
               size    0    age 0
sam-sharefsd   /var/opt/SUNWsamfs/trace/sam-sharefsd
               cust err fatal misc proc debug date
               size    0    age 0
sam-stagerd    /var/opt/SUNWsamfs/trace/sam-stagerd
               cust err fatal misc proc debug date
               size    0    age 0
sam-serverd    /var/opt/SUNWsamfs/trace/sam-serverd
               cust err fatal misc proc debug date
               size    0    age 0
sam-clientd    /var/opt/SUNWsamfs/trace/sam-clientd
               cust err fatal misc proc debug date
               size    0    age 0
sam-mgmt       /var/opt/SUNWsamfs/trace/sam-mgmt
               cust err fatal misc proc debug date
               size    0    age 0
```

(D) - Disk Volume Dictionary

The D display shows the disk volume dictionary, which keeps track of the disk media for disk archiving that has been defined in the `diskvols.conf` file. The dictionary contains information about each VSN, including the capacity, space remaining, and the status flags of the VSN. These flags include unavailable, read only, and bad media.
To invoke this display, type the following command:

```
Command:D
```

Sample Display

This example shows the device configuration display.

Example:***samu D*** Display

```
Disk volume dictionary     samu     5.0 07:48:11 May 8 2008

header
version 460

volumes
magic 340322 version 9 nkeys 2 ndata 2
index    space          capacity       used        flags    volume
    0    12882411520    12887785472    10291200    -----    disk01
    1     6443827200     6443892736       70656    -----    disk02
clients
magic 340322 version 9 nkeys 1 ndata 1
```

Flags

The following table shows the flags for the `D` display.

Flag Values for the ***samu D*** Display

Flag	Description
`l`	Volume is labeled; `seqnum` file has been created. This is set by the administrator to prevent the software from creating a new `seqnum` file.
`r`	Volume is defined on a remote host.
`U`	Volume is unavailable.
`R`	Volume is read only.
`E`	Media error, indicating the software detects a write error on the disk archive directory.

To set or clear a disk volume dictionary flag, use the `diskvols` `samu`(1M) command. See The `diskvols` Command.

(**f**) - File Systems Display

The `f` display shows the components of your Sun QFS file systems.
To invoke this display, type the following command:

```
Command:f
```

Sample Display

This example shows the file systems display.

Example:***samu f*** Display

```
File systems                                     samu 5.0 08:11:24 Sept 8 2008

ty      eq state          device_name   status     high low mountpoint server
ms      10    on                samfs1  m----2----d 90% 70% /sam1
 md     11    on     /dev/dsk/c5t8d0s3
 md     12    on     /dev/dsk/c5t8d0s4
 md     13    on     /dev/dsk/c5t8d0s5
 md     14    on     /dev/dsk/c5t8d0s6
 md     15    on     /dev/dsk/c5t8d0s7
ms      20    on                samfs2  m----2----d 90% 70% /sam2
 md     21    on     /dev/dsk/c5t9d0s3
 md     22    on     /dev/dsk/c5t9d0s4
 md     23    on     /dev/dsk/c5t9d0s5
 md     24    on     /dev/dsk/c5t9d0s6
 md     25    on     /dev/dsk/c5t9d0s7
ma      30    on                  qfs1  m----2----d 90% 70% /qfs1
 mm     31    on    /dev/dsk/c5t10d0s0
 md     32    on    /dev/dsk/c5t10d0s1
ma      40    on                  qfs2  m----2----d 90% 70% /qfs2
 mm     41    on    /dev/dsk/c5t11d0s0
 md     42    on    /dev/dsk/c5t11d0s1
ma      50    on                  qfs3  m----2---r- 90% 70% /qfs3
 mm     51    on    /dev/dsk/c5t12d0s0
 mr     52    on    /dev/dsk/c5t12d0s1
ma      60    on                  qfs4  m----2---r- 90% 70% /qfs4
 mm     61    on    /dev/dsk/c5t13d0s0
 mr     62    on    /dev/dsk/c5t13d0s1
ma     100    on             shareqfs1  m----2c--r- 80% 70% /shareqfs1 spade
 mm    101    on /dev/dsk/c6t50020F2300004655d0s0
 mr    102    on /dev/dsk/c6t50020F2300004655d0s1
ma     110    on             shareqfs2  m----2c--r- 80% 70% /shareqfs2 spade
 mm    111    on /dev/dsk/c6t50020F2300004655d0s6
 mr    112    on /dev/dsk/c6t50020F2300004655d0s7
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) ***f*** Display Field Descriptions

Field	Description
`ty`	Device type.
`eq`	Equipment Number of the device.
`state`	Current operating state of the device. Valid device states are as follows: • on The device is available for access. • `ro` The device is available for read-only access. • `off` The device is not available for access. • down The device is available only for maintenance access. • `idle` The device is not available for new operations. Operations in progress continue until completion. • `nalloc` The `nalloc` flag has been set, which prohibits any future allocation to this device. For more information, see Per-logical unit number (LUN) Allocation Control.
`device_name`	File system name or path to the device.
`status`	Device status. For a description of status codes, see Operator Display Status Codes.
`high`	High disk usage threshold percentage.
`low`	Low disk usage threshold percentage.
`mountpoint`	Mount point of the file system.
`server`	Name of the host system upon which the file system is mounted.

(**F**) - Optical Disk Label Display

The F display shows the label on an optical disk.
To invoke this display, type the following command:

```
Command:F
```

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

(**h**) - Help Display

The h display shows a summary of the samu(1M) displays available. By default, this is the first display that the system presents when you enter the samu(1M) command at the command line.
To invoke this display, type the following command:

```
Command:h
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the *h* Display

Key	Function
Ctrl -b	Page backward
Ctrl -d	Page forward (top portion)
Ctrl -f	Page forward
Ctrl -u	Page backward (top portion)
Ctrl -k	Toggle path display

Sample Display

The online Help has several pages of help screens, but this example shows only the first page of the SAM-QFS configuration. Subsequent help screens show samu(1M) commands.
On a Sun QFS file system, not all of the displays appear in the initial help screen. For example, the removable media displays are not available if you are running a Sun QFS system.

Example: ***samu***(1M) Initial Help Screen for a Sun SAM System

```
Help information    page 1/15    samu 5.0     08:18:13 Sept 8 2008
Displays:
    a    Archiver status                   w    Pending stage queue
    c    Device configuration              C    Memory
    d    Daemon trace controls             D    Disk volume dictionary
    f    File systems                      F    Optical disk label
    h    Help information                  I    Inode
    l    Usage information                 J    Preview shared memory
    m    Mass storage status               K    Kernel statistics
    n    Staging status                    L    Shared memory tables
    o    Optical disk status               M    Shared memory
    p    Removable media load requests     N    File system parameters
    r    Removable media                   P    Active Services
    s    Device status                     R    SAM-Remote
    t    Tape drive status                 S    Sector data
    u    Staging queue                     T    SCSI sense data
    v    Robot catalog                     U    Device table

    more (ctrl-f)
```

(`I`) - Inode Display

The `I` display shows the content of inodes.
You can invoke this display differently, depending on what you need to view, as follows:

- To display inodes for an entire file system, type the command with the following format:

  ```
  Command:I <filesystem>
  ```

  For filesystem, specify the name of a file system.

- To display a specific inode, type the command with the following format:

  ```
  Command:I <inode-number>
  ```

  For inode-number, specify the inode number in either hexadecimal or decimal.

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the *I* Display

Key	Function
Ctrl-b	Previous inode
Ctrl-f	Next inode
Ctrl-k	Advance display format

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This example shows the inode display.

Example:***samu***(1M) ***I*** Display

```
Inode     0x1 (1) format: file          samu     5.0 08:27:14 Sept 8 2008
    incore: y

00008100 mode       -r--------          409cdf57 access_time
00000001 ino        (1)                 1d32ea20
00000001 gen        (1)                 4096b499 modify_time
00000002 parent.ino (2)                 02588660
00000002 parent.gen (2)                 4096b499 change_time
00000000 size_u                         02588660
000c0000 size_l     (786432)            4096b443 creation_time
01000000 rm:media/flags                 409a8a7c attribute_time
00000000 rm:file_offset                 409c0ce6 residence_time
00000000 rm:mau                         00000000 unit/cs/arch/flg
00000000 rm:position                    00000000 ar_flags
00000000 ext_attrs  --------            00000000 stripe/stride/sg
00000000 ext.ino    (0)                 00000000 media  -- --
00000000 ext.gen    (0)                 00000000 media  -- --
00000000 uid        root                00000000 psize      (0)
00000000 gid        root                000000c0 blocks     (192)
00000001 nlink      (1)                 00000600 free_ino   (1536)
00011840 status -n-------  ----- -- --

Extents (4k displayed as 1k):
00_ 000000d0.00 000000e0.00 000000f0.00 00000100.00 00000110.00 00000120.00
06_ 00000130.00 00000140.00 00000150.00 00000160.00 00000170.00 00000180.00
12_ 00000190.00 000001a0.00 000001b0.00 000001c0.00 00000630.00 00000000.00
18_ 00000000.00
```

(J) - Preview Shared Memory Display

The J display shows the shared memory segment for the preview queue.
To invoke this display, type the following command:

```
Command:J
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the *J* Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-u	Half-page backward

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This example shows the preview shared memory display. This sample output has been truncated.

Example: ***samu***(1M) ***J*** Display

```
Preview shared memory   size: 155648    samu 5.0 08:30:05 Sept 8 2008

00000000  00040000 00014d58 00000000 00000000  ......MX........
00000010  00000000 00000000 73616d66 73202d20  ........samfs -
00000020  70726576 69657720 6d656d6f 72792073  preview memory s
00000030  65676d65 6e740000 00026000 00000000  egment..........
00000040  00025fff 00000000 00040000 00014d58  .._...........MX
00000050  00000000 00000000 00000000 00000000  ................
00000060  0000d9e0 00000064 00000000 000001b8  ..Y.....d.......8
00000070  3f800000 447a0000 0000d820 00000008  ?...Dz....X ....
```

(**K**) - Kernel Statistics Display

The K display shows kernel statistics, such as the number of inodes currently in memory.
To invoke this display, type the following command:

```
Command:K
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the ***K*** Display

Key	Function
Ctrl-b	Page backward
Ctrl-f	Page forward

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This example shows the kernel statistics display.

Example: ***samu***(1M) ***K*** Display

```
Kernel statistics                      samu 5.0     08:33:19 Sept 8 2008

module: sam-qfs  name: general instance: 0 class: fs
version                        4.4.sam-qfs, gumball 2004-05-07 12:12:04
configured file systems       8
mounted file systems           8
nhino                          16384
ninodes                        129526
inocount                       129527
inofree                        128577
```

(**l**) - Usage Display

The `l` display shows the usage information for the file system, including the capacity and space used for each library and file system.
To invoke this display, type the following command:

```
Command:l
```

Sample Display

This example shows an example of a usage display.

Example:***samu***(1M) ***l*** Display

```
Usage information                     samu     5.0 08:36:27 Sept 8 2008

hostid = 80e69e6e OS name: SunOS Architecture: SPARC CPUs: 2 (2 online)

library     40: capacity    389.3G     bytes space    291.1G    bytes, usage     25%
library     51: capacity    9.5G       bytes space    9.5G      bytes, usage     0%
library     55: capacity    0          bytes space    0         bytes, usage     0%
library     56: capacity    10.7G      bytes space    10.7G     bytes, usage     0%
library totals: capacity    409.5G     bytes space    311.3G    bytes, usage     24%

filesystem samfs3: capacity    54.5M    bytes space    13.4M    bytes, usage     75%
filesystem samfs4: capacity    319.5M   bytes space    298.0M   bytes, usage      7%
filesystem samfs7: capacity    96.6M    bytes space    69.6M    bytes, usage     28%
filesystem samfs6: capacity    5.0G     bytes space    4.9G     bytes, usage      3%
filesystem samfs8: capacity    5.0G     bytes space    4.9G     bytes, usage      2%
filesystem totals: capacity    10.5G    bytes space    10.2G    bytes, usage      3%
```

Note -
In versions of the software before 4.3, this display showed license information for the file system.

(**L**) - Shared Memory Tables

The L display shows the location of the shared memory tables. It also shows some system defaults that are kept in shared memory.
To invoke this display, type the following command:

```
Command:L
```

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This shows the shared memory tables.

Example: ***samu***(1M) ***L*** Display

```
Shared memory tables                    samu 5.0 08:38:31 May  8 2008

shm ptr tbl:                            defaults:
size            12000 (73728)           optical         mo
left            44c8 (17608)            tape            lt
scanner pid     1861                    timeout         600
fifo path       01b0 /var/opt/SUNWsamfs/previews        100
dev_table       01cc                    stages          1000
first_dev       0450                    log_facility    184
scan_mess       cf50                    dio minfilesize 100
preview_shmid   1                       label barcode   FALSE
flags           0x20000000              barcodes low    FALSE
preview stages  55776                   export unavail  FALSE
preview avail   100                     attended        TRUE
preview count   0                       start rpc       FALSE
preview sequence  445
age factor         1                    vsn factor      1000
fs tbl ptr 0xd820                       fs count  8
fseq  10 samfs1 state 0      0      0      0      0
fseq  20 samfs2 state 0      0      0      0      0
fseq  30 qfs1 state 0      0      0      0      0
fseq  40 qfs2 state 0      0      0      0      0
fseq  50 qfs3 state 0      0      0      0      0
fseq  60 qfs4 state 0      0      0      0      0
fseq  100 shareqfs1 state 0      0      0      0      0
fseq  110 shareqfs2 state 0      0      0      0      0
```

(m) - Mass Storage Status Display

The m display shows the status of mass storage file systems and their member drives. This display shows only mounted file systems. To invoke this display, type the following command:

```
Command:m
```

Sample Display

This example shows the m display. Member drives are indented one space and appear directly below the file system to which they belong.

Example: *samu*(1M) *m* Display

```
Mass storage status                      samu 5.0     08:41:11 Sept 8 2008

ty       eq  status        use state ord  capacity      free    ra  part high low
ms       10  m----2----d    1% on          68.354G   68.343G    1M    16  90% 70%
 md       11                1% on      0   13.669G   13.666G
 md       12                1% on      1   13.669G   13.667G
 md       13                1% on      2   13.669G   13.667G
 md       14                1% on      3   13.674G   13.672G
 md       15                1% on      4   13.674G   13.672G
ms       20  m----2----d    1% on          68.354G   68.344G    1M    16  90% 70%
 md       21                1% on      0   13.669G   13.667G
 md       22                1% on      1   13.669G   13.667G
 md       23                1% on      2   13.669G   13.667G
 md       24                1% on      3   13.674G   13.672G
 md       25                1% on      4   13.674G   13.672G
ma       30  m----2----d    4% on          64.351G   61.917G    1M    16  90% 70%
 mm       31                1% on      0    4.003G    3.988G [8363840 inodes]
 md       32                4% on      1   64.351G   61.917G
ma       40  m----2----d    1% on          64.351G   64.333G    1M    16  90% 70%
 mm       41                1% on      0    4.003G    3.997G [8382784 inodes]
 md       42                1% on      1   64.351G   64.333G
ma       50  m----2---r-    1% on          64.351G   64.333G    1M    16  90% 70%
 mm       51                1% on      0    4.003G    3.997G [8382784 inodes]
 mr       52                1% on      1   64.351G   64.333G
ma       60  m----2---r-    1% on          64.351G   64.331G    1M    16  90% 70%
 mm       61                1% on      0    4.003G    3.997G [8382784 inodes]
 mr       62                1% on      1   64.351G   64.331G
ma      100  m----2c--r-    2% on         270.672G  265.105G    1M    16  80% 70%
 mm      101                1% on      0    2.000G    1.988G [4168992 inodes]
 mr      102                2% on      1  270.672G  265.469G
ma      110  m----2c--r-    3% on         270.656G  263.382G    1M    16  80% 70%
 mm      111                1% on      0    2.000G    1.987G [4167616 inodes]
 mr      112                2% on      1  270.656G  264.736G
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) ***m*** Display Field Descriptions

Field	Description
`ty`	Device type.
`eq`	Equipment Number of the mass storage device.
`status`	Device status. For a description of status codes, see Operator Display Status Codes.
`use`	Percentage of disk space in use.
`state`	Current operating state of the mass storage device.
`ord`	Ordinal number of the disk device within the storage Family Set.
`capacity`	Number of 1024-byte blocks of usable space on the disk.
`free`	Number of 1024-byte blocks of disk space available.
`ra`	Read-ahead size in kilobytes.
`part`	Partial stage size in kilobytes.
`high`	High disk usage threshold percentage.
`low`	Low disk usage threshold percentage.

(**M**) - Shared Memory Display

The M display shows the raw shared memory segment in hexadecimal. This is a device table.
To invoke this display, type the following command:

```
Command:M
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the *M* Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-u	Half-page backward

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This example shows the shared memory display. The sample output has been truncated.

Example:*samu*(1M) *M* Display

```
Shared memory       size: 73728         samu 5.0     08:43:20 May  8 2008

00000000  00040000 00014d58 00000000 00000000  ......MX........
00000010  00000000 00000000 73616d66 73202d20  ........samfs -
00000020  73686172 6564206d 656d6f72 79207365  shared memory se
00000030  676d656e 74000000 00012000 000044c8  gment..... ...DH
00000040  0000dd20 00000000 00000742 00000745  ..] .......B...E
00000050  00000001 00000000 00000000 c0000000  ............@...
00000060  00000001 0001534d 00000000 00000000  ......SM........
00000070  00000000 00000000 00000000 00000000  ................

00000080  00000000 00000000 00000000 00000000  ................
00000090  20000000 000001b0 000001cc 00000450   ......0...L...P
000000a0  0000cf50 00000001 00000001 4c696365  ..OP........Lice
000000b0  6e73653a 204c6963 656e7365 206e6576  nse: License nev
000000c0  65722065 78706972 65732e00 00000000  er expires......
000000d0  00000000 00000000 00000000 00000000  ................
000000e0  00000000 00000000 00000000 00000000  ................
000000f0  00000000 00000000 00000000 00000000  ................
```

(**n**) - Staging Status Display

The n display shows the status of the stager for all media. It displays a list of outstanding stage requests.

- To display the staging status for all staging activity, type the command with the following format:

  ```
  Command:n
  ```

- To display the staging status for a specific media type, type the command with the following format:

```
Command:n <mt>
```

For mt, specify one of the media types shown in the mcf(4) man page.

Sample Display

This example shows the staging status display.

Example:***samu***(1M) ***n*** Display

```
Staging status                          samu 5.0     08:47:16 May  8 2008

Log output to: /var/opt/SUNWsamfs/stager/log

Stage request: dt.DAT001
Loading VSN DAT001

Staging queues
ty pid    user         status     wait  files  vsn
dt 16097  root         active     0:00    12   DAT001
```

(**N**) - File System Parameters Display

The N display shows all mount point parameters, the superblock version, and other file system information.
To invoke this display, type the following command:

```
Command:N
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the ***N*** Display

Key	Function
Ctrl -b	Previous file system
Ctrl -d	Page partitions forward
Ctrl -f	Next file system
Ctrl -i	Detailed status interpretations
Ctrl -u	Page partitions backward

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This example shows the file system parameters display.

Example:***samu***(1M) ***N*** Display

```
File system parameters                  samu 5.0     08:55:19 Sept 8 2008

mount_point    : /sam1                  partial         : 16k
fs_type        : 6                      maxpartial      : 16k
server         :                        partial_stage   : 16384
filesystem name: samfs1                 flush_behind    : 0
eq  type       : 10 ms                  stage_flush_beh: 0
state version  :    0    2              stage_n_window : 262144
(fs,mm)_count  :    5    0              stage_retries   : 3
sync_meta      : 0                      stage timeout   : 0
stripe         : 0                      dio_consec r,w :    0    0
mm_stripe      : 1                      dio_frm_min r,w:  256  256
high  low      :  90%  70%              dio_ill_min r,w:    0    0
readahead      : 1048576                ext_bsize       : 4096
writebehind    : 524288
wr_throttle    : 16777216
rd_ino_buf_size: 16384
wr_ino_buf_size: 512
config         : 0x08520530             mflag           : 0x00000044
status         : 0x00000001

Device configuration:
ty   eq state   device_name                        fs family_set
md   11 on      /dev/dsk/c5t8d0s3                  10 samfs1
md   12 on      /dev/dsk/c5t8d0s4                  10 samfs1
md   13 on      /dev/dsk/c5t8d0s5                  10 samfs1
md   14 on      /dev/dsk/c5t8d0s6                  10 samfs1
md   15 on      /dev/dsk/c5t8d0s7                  10 samfs1
```

(**o**) - Optical Disk Status Display

The o display shows the status of all optical disk drives configured within the environment.
To invoke this display, type the following command:

```
Command:o
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the *o* Display

Key	Function
Ctrl-b	Page backward
Ctrl-f	Page forward
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-k	Select (manual, automated library, both, priority)
Ctrl-u	Half-page backward

Sample Display

This example shows the optical disk status display.

Example: ***samu***(1M) ***o*** Display

```
Optical disk status            samu    5.0     Thu Oct 11 13:15:40
ty  eq  status      act  use  state  vsn
mo  35  --l---wo-r    1  29%  ready  oper2
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) ***o*** Display Field Descriptions

Field	Description
`ty`	Device type.
`eq`	Equipment Number of the optical disk.
`status`	Device status. For a description of status codes, see Operator Display Status Codes.
`act`	Activity count.
`use`	Percentage of cartridge space used.
`state`	Current operating state of the optical disk. Valid device states are as follows: • `ready` The device is on, and the disk is loaded in the transport; available for access. • `notrdy` The device is on, but no disk is present in the transport. • `idle` The device is not available for new connections. Operations in progress continue until completion. • `off` The device is not available for access. • down The device is available only for maintenance access.
`vsn`	Volume Serial Name assigned to the optical disk, or the keyword `nolabel` if the volume is not labeled.

(**p**) - Removable Media Load Requests Display

The p display lists information about pending load requests for removable media. You can use the mt argument to select either a specific type of media, such as DLT tape, or a family of media, such as tape. The priority display lists the priority in the preview queue, rather than the user queue, and sorts the entries by priority.
It displays mount requests in the following formats:

- Both manual and automated library requests by user
- Both manual and automated library requests by priority
- Manual requests only
- Automated library requests only

You can invoke this display differently, depending on what you need to view, as follows:

- To display mount requests for all removable devices currently selected, type the command with the following format:

  ```
  Command:p
  ```

- To display mount requests for devices of a given removable media type, type the command with the following format:

  ```
  Command:p <mt>
  ```

  For mt, specify one of the media types shown in the `mcf`(4) man page.

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the ***p*** Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-k	Toggle between the different display formats
Ctrl-u	Half-page backward

Sample Display

This example shows the removable media load requests display.

Example: ***samu***(1M) ***p*** Display

```
Removable media load requests all both  samu 5.0 09:14:19 Sept 8 2008
    count: 1

index type pid     user         rb   flags     wait count  vsn
   0  dt   15533   root         150  W--f---   0:00        DAT001
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) ***p*** Display Field Descriptions

Field	Description
`index`	Index number in the preview table.
`type`	Device type code assigned to the removable media.
`pid`	UNIX process identifier. A process identifier of 1 indicates NFS access.
`user`	Name assigned to the user requesting the load.
`priority`	Priority of the request.
`rb`	Equipment Number of the automated library in which the requested VSN resides.
`flags`	Flags for the device. See the Flag table.
`wait`	The elapsed time since the mount request was received.
`count`	The number of requests for this VSN, if it is a stage.
`vsn`	Volume serial name of the volume.

Flags

The following table shows the flags for the p display.

Flags Field for the ***samu***(1M) ***p*** Display

Flag	Description
W-------	Write access requested.
}}{{{}b----	Entry is busy.
-c---	Clear VSN requested.
--f--	File system requested.
---N-	Media is foreign to the file system.
---S{-}	Flip side already mounted.
-------s	Stage request flag.

(**P**) - Active Services Display

The P display lists the services registered with the Sun QFS single port multiplexer.
To invoke this display, type the following command:

```
Command:P
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the ***P*** Display

Key	Function
Ctrl-b	Page backward
Ctrl-f	Page forward

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This example shows the active services display.

Example: ***samu***(1M) ***P*** Display

```
Active Services                         samu     5.0    09:08:33 Sept 8 2008

Registered services for host &rdquo;pup&rsquo;:
    sharedfs.qfs2
    sharedfs.qfs1
  2 services registered.
```

(**r**) - Removable Media Status Display

The r display monitors the activity on removable media devices such as tape drives. You can monitor either a specific type of device, such as video tape, or a family of devices such as all tape devices.

- To display the status for all removable media devices, type the command with the following format:

```
Command:r
```

- To display the status for a specific device, type the command with the following format:

```
Command:r <eq>
```

For eq, specify the Equipment Number for the device.

Sample Display

This example shows the removable media status display.

Example: ***samu***(1M) ***r*** Display

```
Removable media status: all            samu 5.0     09:11:27 Sept 8 2008

ty   eq  status      act  use  state    vsn
dt  150  --l------r    0  63%  ready    DAT001
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) ***r*** Display Field Descriptions

Field	Description
`ty`	Device type.
`eq`	Equipment Number of the drive.
`status`	Device status. For a description of status codes, see Operator Display Status Codes.
`act`	Activity count.
`use`	Percentage of cartridge space used.
`state`	Current operating state of the removable media. Valid device states are as follows: • `ready` The device is on, and the disk or tape is loaded in the transport; available for access. • `notrdy` The device is on, but no disk or tape is present in the transport. • `idle` The device is not available for new connections. Operations in progress continue until completion. • `off` The device is not available for access. • down The device is available only for maintenance access. • `nalloc` The `nalloc` flag has been set, which prohibits any future allocation to this device. For more information, see Per-logical unit number (LUN) Allocation Control.
`vsn`	Volume serial name assigned to the volume, or the keyword `nolabel` if the volume is not labeled. Blank if no volume is present in the transport, or device is off.

(**R**) - Sun SAM-Remote Information Display

The R display shows information and status on Sun SAM-Remote configurations.
To invoke this display, type the following command:

```
Command:R
```

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

(**s**) - Device Status Display

The s display shows the status for all devices configured within the environment.
To invoke this display, type the following command:

```
Command:s
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the *s* Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-u	Half-page backward

Sample Display

This example shows the device status display.

Example: *samu*(1M) *s* Display

```
Device status                          samu     5.0     09:14:05 Sept 8 2008

ty     eq     state    device_name                     fs  status       pos
sk    100     on       /etc/opt/SUNWsamfs/dcstkconf    100 m--------r   stk_dismount(2275) 0,
volser 700073
sg    120     on       /dev/rmt/2cbn                   100 ---------p   empty
sg    130     on       /dev/rmt/5cbn                   100 --l----o-r   Ready for data transfer
sg    140     on       /dev/rmt/6cbn                   100 ---------p   empty
sg    150     on       /dev/rmt/4cbn                   100 ---------p   empty
hy    151     on       historian                       151 ----------
```

Field Descriptions

The following table shows the field descriptions for this display.

{samu}}(1M) *s* Display Field Descriptions

Field	Description
ty	Device type.
eq	Equipment Number of the device.
state	Current operating state of the device.
device_name	Path to the device. For file system devices, this is the file system name.
fs	Equipment Number of the family, set to which the device belongs.
status	Device status. For a description of status codes, see Operator Display Status Codes.

(S) - Sector Data Display

The S display shows raw device data.
To invoke this display, type the following command:

```
Command:S
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the ***S*** Display

Key	Function
Ctrl-b	Previous sector
Ctrl-d	Page forward (top portion)
Ctrl-f	Next sector
Ctrl-k	Advance display format
Ctrl-u	Page backward (top portion)

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

(**t**) - Tape Drive Status Display

The t display shows the status of all tape drives configured within the environment.
To invoke this display, type the following command:

```
Command:t
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the ***t*** Display

Key	Function
Ctrl-b	Page backward
Ctrl-f	Page forward

Sample Display

This example shows the tape drive status display.

Example:***samu***(1M) ***t*** Display

```
Tape drive status                       samu     5.0     09:21:07 Sept 8 2008

ty   eq  status      act  use  state    vsn
sg  120  ---------p    0   0%  notrdy          empty
sg  130  ---------p    0   0%  notrdy          empty
sg  140  ---------p    0   0%  notrdy          empty
sg  150  --l------r    0  41%  ready    700088 idle
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) t Display Field Descriptions

Field	Description
`ty`	Device type.
`eq`	Equipment Number of the drive.
`status`	Device status. For a description of status codes, see Operator Display Status Codes.
`act`	Activity count.
`use`	Percentage of cartridge space used.
`state`	Current operating state of the removable media. Valid device states are as follows: • `ready` The device is on and the disk or tape is loaded in the transport; available for access. • `notrdy` The device is on but no disk or tape is present in the transport. • `idle` The device is not available for new connections. Operations in progress continue until completion. • `off` The device is not available for access. • down The device is available only for maintenance access. • `nalloc` The `nalloc` flag has been set, which prohibits any future allocation to this device. For more information, see Per-logical unit number (LUN) Allocation Control.
`vsn`	Volume serial name assigned to the volume, or the keyword nolabel if volume is not labeled. Blank if no volume is present in the transport, or device is off.

(T) - SCSI Sense Data Display

The T display shows the SCSI status of a SCSI device.
To invoke this display, type the following command:

```
Command:T
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the T Display

Key	Function
Ctrl-b	Previous equipment
Ctrl-f	Next equipment

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

(u) - Staging Queue Display

The u display lists all files in the staging queue.
To invoke this display, type the following command:

```
Command:u
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the *u* Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-k	Display the path on the second line of each entry
Ctrl-u	Half-page backward

Sample Display

This example shows the staging queue display.

Example: ***samu***(1M) ***u*** Display

```
Staging queue by media type: all        samu 5.0     09:24:23 Sept 8 2008
     volumes 1 files 22

ty     length  fseq        ino   position     offset vsn

dt   451.611k    20       1030      207cc        473 DAT001
dt   341.676k    20       1031      207cc        7fc DAT001
dt   419.861k    20       1032      207cc        aa9 DAT001
dt   384.760k    20       1033      207cc        df2 DAT001
dt   263.475k    20       1034      207cc       10f5 DAT001
dt   452.901k    20       1035      207cc       1305 DAT001
dt   404.598k    20       1036      207cc       1690 DAT001
dt   292.454k    20       1037      207cc       19bb DAT001
dt   257.835k    20       1038      207cc       1c05 DAT001
dt   399.882k    20       1040      207cc       1e0b DAT001
dt   399.882k    40       1029      208d7          2 DAT001
dt   257.835k    40       1030      208d7        323 DAT001
dt   292.454k    40       1031      208d7        528 DAT001
dt   404.598k    40       1032      208d7        772 DAT001
dt   452.901k    40       1033      208d7        a9d DAT001
dt   263.475k    40       1034      208d7        e28 DAT001
dt   384.760k    40       1035      208d7       1038 DAT001
dt   419.861k    40       1036      208d7       133b DAT001
dt   341.676k    40       1037      208d7       1684 DAT001
dt   451.611k    40       1038      208d7       1931 DAT001
dt   161.326k    40       1039      208d7       1cba DAT001
dt   406.400k    40       1040      208d7       1dfe DAT001
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) ***u*** Display Field Descriptions

Field	Description
`ty`	Device type.
`length`	File length.

`fseq`	File system equipment number.
`ino`	The inode number.
`position`	The position of the archive file on the specific medium.
`offset`	Offset of the archive file on the specific medium.
`vsn`	Volume serial name of the volume.

(**U**) - Device Table Display

The U display shows the device table in a human-readable form.

- To display the device table for all devices, type the command with the following format:

```
Command:U
```

- To display the device table for a specific device, type the command with the following format:

```
Command:U <eq>
```

  For eq, specify the Equipment Number of the device.

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the *U* Display

Key	Function
Ctrl-b	Previous equipment
Ctrl-f	Next equipment

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This example shows the device table display.

Example:*samu*(1M) *U* Display

```
Device table: eq: 10    addr: 00000450  samu 5.0     09:28:40 Sept 8 2008

message:

0004000000014d58 0000000000000000       00000000 delay
0000000000000000 mutex                  00000000 unload_delay
00000aa8 next
73616d66 set:  samfs1
73310000
00000000
00000000
000a000a eq/fseq
08010801 type/equ_type
0000     state
00000000 st_rdev
00000000 ord/model
00000000 mode_sense
00000000 sense
00000000 space
00000000 capacity
00000000 active
00000000 open
00000000 sector_size
00000000 label_address
00000000 vsn:
00000000
00000000
00000000
00000000 status: ----------
00000000 dt
73616d66 name: samfs1
```

(**v**) - Automated Library Catalog Display

The v display shows the location and VSN of all disks or tapes currently cataloged in the automated library.

- To display the catalog for all devices, type the command with the following format:

```
Command:v
```

- To display catalog information for a specific device, type the command with the following format:

```
Command:v _eq_
```

For eq, specify the Equipment Number of the device. Type the keyword `historian` to view the historian catalog. For a list of all device names and Equipment Numbers, see (c) - Device Configuration Display.

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the *v* Display

Key	Function
Ctrl -b	Page backward.
Ctrl -d	Next library catalog.

Ctrl -f	Page forward.
Ctrl -i	Detailed, two-line display format. When you enter Ctrl -i once, it shows times and barcodes. When you enter Ctrl -i a second time, it shows volume reservations on the second line.
Ctrl -k	Advance sort key. After you enter Ctrl-k, you can enter one of the following to select a sort key:
	1 - sort by slot.
	2 - sort by count.
	3 - sort by usage.
	4 - sort by VSN.
	5 - sort by access time.
	6 - sort by barcode.
	7 - sort by label time.
Ctrl-u	Previous automated library catalog.
/	Search for VSN.
%	Search for barcode.
$	Search for slot.

Sample Display

This example shows the automated library catalog display.

Example: ***samu***(1M) ***v*** Display

```
Robot VSN catalog by slot       : eq 100samu     5.0     09:30:25 Sept 8 2008
    count 32
slot          access time count use flags         ty vsn

   0     2004/05/08 08:35   64   0% -il-o-b-----  sg 700071
   1     2004/05/08 09:08   27  12% -il-o-b-----  sg 700073
   2     2004/05/08 09:12   26  12% -il-o-b-----  sg 700077
   3     2004/05/08 08:39   37  40% -il-o-b-----  sg 700079
   4     2004/05/08 09:16   24   6% -il-o-b-----  sg 700084
   5     2004/05/08 09:18   24  41% -il-o-b-----  sg 700088
   6     none                0   0% -il-o-b-----  sg 700090
   7     none                0   0% -il-o-b-----  sg 700092
   8     none                0   0% -il-o-b-----  sg 000155
   9     none                0   0% -il-o-b-----  sg 000156
  10     none                0   0% -il-o-b-----  sg 000157
  11     none                0   0% -il-o-b-----  sg 000158
  12     none                0   0% -il-o-b-----  sg 000154
  13     none                0   0% -il-o-b-----  sg 000153
  14     none                0   0% -il-o-b-----  sg 000152
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) ***v*** Display Field Descriptions

Field	Description
`Robot VSN catalog`	Name of the specified automated library and time the display refreshed.

`count`	Number of slots allocated in this library's catalog.
`slot`	Slot number within the specified library.
`access time`	Time the volume was last accessed.
`count`	Number of accesses to this volume since the last audit operation.
`use`	Percentage of space used for the volume.
`flags`	Flags for the device. See the flag table for information about the flags.
`ty`	Device type.
`vsn`	Volume serial name of the volume.

Flags

The following table shows the flags using in the `flags` field. In some cases, more than one flag can occur in a field, and one flag overrides the other.

Flags Field for ***samu***(1M) ***v*** Display

Flags	Description
A-----------	Volume needs audit.
}}{{{}I---------	Slot in use.
-l--------	Labeled. Overrides `N`.
-N--------	Unlabeled. This volume is foreign to the environment.
--E-------	Media error. Set when the software detects a write error on a cartridge.
--o-----	Slot occupied.
---C----	Volume is a cleaning tape. Overrides p.
---p----	Priority VSN.
----b---	Barcode detected.
-----W--	Write protect. Set when the physical write protection mechanism is enabled on a cartridge.
------R{-}--	Read only.
-------c{-}-	Recycle.
--------d{-}	Duplicate VSN. Overrides `U`.
--------U{-}	Volume unavailable.
-----------f	Archiver found volume full.
-----------X	Export slot.

(w) - Pending Stage Queue

The `w` display shows queued stage requests for which the volumes have not yet been loaded.

- To display the pending stage queue for all media, type the command with the following format:

```
Command:w
```

- To display the pending stage queue for a specific media type, type the command with the following format:

```
Command:w <mt>
```

For mt, specify one of the media types shown in the `mcf`(4) man page.

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the *w* Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-k	Display the path on the second line of each entry
Ctrl-u	Half-page backward

Sample Display

This example shows the pending stage queue.

Example:*samu*(1M) *w* Display

```
Pending stage queue by media type: all     samu     5.0     Thu Oct 11 13:20:27
    volumes 1 files 13

ty     length  fseq  ino  position  offset  vsn

at     1.383M     1   42     3a786    271b  000002
at     1.479M     1   56     3a786    5139  000002
at   1018.406k    1   60     3a786    6550  000002
at     1.000M     1   65     3a786    7475  000002
at     1.528M     1   80     3a786    99be  000002
at     1.763M     1   92     3a786    ce57  000002
at     1.749M     1  123     3a786   11ece  000002
at    556.559k    1  157     3a786   1532f  000002
at    658.970k    1  186     3a786   17705  000002
at    863.380k    1  251     3a786   1dd58  000002
at     1.268M     1  281     3a786   1f2b7  000002
at     1.797M     1  324     3a786   23dfa  000002
at     1.144M     1  401     3a786   2bb6d  000002
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) *w* Display Field Descriptions

Field	Description
`ty`	Device type.
`length`	File length.
`fseq`	File system Equipment Number.
`ino`	The inode number.

`position`	The position (in decimal format) of the archive file on the specific medium.
`offset`	Offset of the archive file on the specific medium.
`vsn`	Volume serial name of the volume.

Status Codes

The operator displays have different status codes for the removable media device displays and the file system displays. The following sections describe these status codes.

Removable Media Device Display Status Codes

The `o`, `r`, `s`, and `t` operator displays show status codes for removable media devices. Status codes are displayed in a 10-position format, reading from left (position 1) to right (position 10).
These status codes do not apply to the `samu(1M)` `f`, `m`, and `v` displays. For information about the status codes for the `f` and `m` displays, see File System Display Status Codes. For information about the status codes for the `v` display, see (v) - Automated Library Catalog Display.

The following table defines the status codes for each position.

Status Code for Removable Media Device Display

Status Bit	Meaning for a Device
`s---------`	Media is being scanned.
`m---------`	The automated library is operational.
`M---------`	Maintenance mode.
}}{{{}E-------	Device received an unrecoverable error in scanning.
}}{{{}a-------	Device is in audit mode.
-l------	Media has a label.
-N------	Foreign media.
-L------	Media is being labeled.
--I-----	Waiting for device to idle.
--A-----	Needs operator attention.
-=C=--	Needs cleaning.
-=U=--	Unload has been requested.
--=R=-	Device is reserved.
---=w{-}--	A process is writing on the media.
----=o{-}-	Device is open.
-----=P{-}	Device is positioning (tape only).
-----=F{-}	For automated libraries, all storage slots occupied. For tape and magneto-optical drives, media is full.
`---------R`	Device is ready and the media is read-only.
`---------r`	Device is spun up and ready.
`---------p`	Device is present.
`---------W`	Device is write protected.

File System Display Status Codes

The f and m operator displays show status codes for file systems. Status codes are displayed in an 11-position format, reading from left (position 1) to right (position 11).
These status codes do not apply to the samu(1M) c, o, r, s, t, or v displays. For information about the status codes for the o, r, s, and t displays, see Status Codes for Removable Media Device Display . For information about the status codes for the v display, see (v) - Automated Library Catalog Display.

The following table defines the status codes for each position.

File System Display Status Codes

Status Bit	Meaning for a File System
m----------	File system is currently mounted.
M----------	File system is being mounted.
}}{{{}u--------	File system is being unmounted.
-A-------	File system data is being archived.
--R------	File system data is being released.
-S----	File system data is being staged.
--1---	Sun SAM file system version 1.
--2---	Sun SAM file system version 2.
---c--	Sun QFS shared file system.
----W{-}--	Single writer.
-----R{-}-	Multireader.
------r{-}	mr devices.
----------d	md devices.

States of an Operator Display Device

The c, m, o, r, s, and t operator displays show state codes for each device. These codes represent the current access state for the device. The following table defines the state codes.

Operator Display Device States

Device State	Description
on	The device is available for access. For certain displays, this state might be superseded by the states ready or notrdy.
ro	The device is available for read-only access. For certain displays, this state might be superseded by the states ready or notrdy.
off	The device is not available for access. For tape and optical disk drives, possible reasons for the device to be in the off state include the following:\ • Cleaning was requested, but no cleaning cartridge was found in the automated library. • The cleaning cartridge cannot be loaded or unloaded from the drive. • Initialization found the drive status to be full, and attempts to clear the drive failed. • The system was unable to clear a cartridge from a drive. • Opening the drive for I/O failed during spin-up. • An error other than NOT READY was received when spinning down the drive for unloading. • Opening the standard tape driver on the drive failed during spin-up.
down	The device is available for maintenance access only.
idle	The device is not available for new connections. Operations in progress continue until completion.

ready	The device is on and the disk or tape loaded in the transport is available for access.
notrdy	The device is on, but no disk or tape is present in the transport.
unavail	The device is unavailable for access and cannot be used for automatic operations. You can continue to use the load(1M) and unload(1M) commands for moving media while the device is in the unavail state.
nalloc	The nalloc flag has been set, which prohibits any future allocation to this device. For more information, see Per-logical unit number (LUN) Allocation Control.

You can use the samu(1M) down, off, and on device state commands to change device states. You can enter these commands from any samu (1M) display, but if you enter them from the c, m, o, r, s, or t display, you can see the change in the display.
The following procedures show how to change a device's state from down to on and from on to down.

To Change a Drive State From **down** to **on**

1. Enter one of the following character keys to show a samu(1M) display with drive and automated library device states: c, m, o, r, s, or t.
2. Choose a device and verify that it is in the down state.
3. Use the :off command to stop all activity for the device.

```
Command:off <eq>
```

   where eq is the Equipment Number of the device.

4. Use the :on command.
   For example:

```
Command:on <eq>
```

   For eq, specify the Equipment Number of the device.

Operator Commands

The following topics describe the operator commands that you can enter from the samu(1M) operator utility's command interface. You can enter the commands from any display.

The following types of operator commands are available:

- Device Commands
- File System Commands: I/O Management
- File System Commands: Direct I/O Management
- File System Commands: Sun QFS Shared File Systems
- File System Commands: Miscellaneous
- Miscellaneous Commands
  If you want to enter any operator commands from the Solaris OS command line, you must use them as arguments to the samcmd(1M) command. For more information about the samcmd(1M) command, see the samcmd(1M) man page.

In the following subsections, each samu(1M) command is prefaced with a colon (:) character to indicate the entry is a a command and not a series of hot keys.

Device Commands

The following table shows the device commands and their actions.

Device Command Actions

Command	Action
down	Terminates operation on device eq.

idle	Restricts access to device eq by preventing new connections to the device. Existing operations continue until completion.
off	Logically turns off device eq.
on	Logically turns on device eq.
unavail	Selects device eq and makes it unavailable for use with the file system. You might set a drive state to `unavail`, for example, in a disaster recovery situation in which you are trying to load media to restore a file system and you do not want the Sun SAM software to attempt to use this drive.
unload	Unloads the mounted media for the specified removable media device eq. For magazine devices, the unload command unloads the mounted cartridge and ejects the magazine.
`nalloc`	Sets the `nalloc` flag on the device, which prohibits any allocation to this device. For more information, see Per-logical unit number (LUN) Allocation Control.
`alloc`	Removes the `nalloc` flag from the device. The `nalloc` flag prohibits any allocation to this device. The on command also removes this flag. For more information, see Per-logical unit number (LUN) Allocation Control.

All these commands use the following syntax:

```
:<command> <eq>
```

For eq, specify the Equipment Number of the device.

File System Commands: I/O Management

The following commands enable you to manage I/O characteristics dynamically.

The **`flush_behind`** Command

The `flush_behind` command sets the maximum `flush_behind` value. When set to a value greater than 0, modified pages that are being written sequentially are written to disk asynchronously to help the Solaris kernel layer keep the pages clean. By default, the maximum value is 0, which disables `flush_behind`.

```
:flush_behind <eq> <value>
```

For value, specify an integer number of kilobytes such that `0`≤ value≤ `8192`.
For eq, specify the Equipment Number for the file system.

The **`force_nfs_async`** and **`noforce_nfs_async`** Commands

These commands enable you to control whether the file system caches NFS data written to the server even if NFS has requested that the data be written synchronously through to disk. The `force_nfs_async` command caches NFS data.
The `force_nfs_async` command is effective only if the file system is mounted as an NFS server and only if the clients are mounted with the `noac` NFS mount option. For more information about mounting an NFS file system, see the `mount_nfs`(1M) man page.
The `noforce_nfs_async` command, which is the default, synchronously writes data through to disk.

```
:force_nfs_async <eq>
:noforce_nfs_async <eq>
```

For eq, specify the Equipment Number for the file system.

> Caution -
> The `force_nfs_async` option violates NFS protocols. Use this command with caution. In the event of a server interruption, data can be lost. Data is cached on the NFS server and cannot be seen immediately by all the clients if there are multiple NFS servers. Multiple NFS servers can be enabled within the Sun QFS shared file system. For more information about the Sun QFS shared file system, see Configuring a Sun QFS Shared File System.

The **readahead** Command

The readahead command specifies the maximum number of bytes that can be read ahead by the file system. The default contig number is 8 (131072 bytes).

```
:readahead <eq> <contig>
```

For eq, specify the Equipment Number for the file system.
For contig, specify units of 1-kilobyte blocks. This must be an integer such that 1 < contig < 8192. The contig specified is truncated to a multiple of 8 kilobytes.

For example, the following command sets the maximum contiguous block size to 262,144 bytes for the file system defined as Equipment Number 3:

```
:readahead 3 256
```

This value can also be configured in the samfs.cmd file by specifying the readahead directive. For more information, see the samfs.cmd(4) man page.

The **sw_raid** and **nosw_raid** Commands

These commands specify whether the file system aligns the writebehind buffer. Specify sw_raid if the software RAID feature of a package such as Solstice DiskSuite™ is also used on this file system. The default setting is nosw_raid.

```
:sw_raid <eq>
:nosw_raid <eq>
```

For eq, specify the Equipment Number for a file system.

The **writebehind** Command

The writebehind command specifies the maximum number of bytes that can be written behind by a file system. The default contig number is 8 (131072 bytes).

```
:writebehind <eq> <contig>
```

For eq, specify the Equipment Number for a file system.
For contig, specify units of 1-kilobyte blocks. This must be an integer such that 1 < contig < 8192.

For example, the following command sets the maximum contiguous block size to 262,144 bytes for the file system defined as Equipment Number 50:

```
:writebehind 50 256
```

This value can also be configured in the samfs.cmd file by specifying the writebehind directive. For more information, see the samfs.cmd (4) man page.

The **wr_throttle** Command

The wr_throttle command sets the number of outstanding write kilobytes for one file. The default is the number of kilobytes that is 2% of the memory size.
If the percentage cannot be calculated, set the value to 100 MBytes. To determine memory size on Solaris:

```
npages = sysconf(_SC_PHYS_PAGES);

pagesizeb = sysconf(_SC_PAGESIZE);

fprintf(fp, "Memory size:\t\t%.1f MBytes (%.1f M2Bytes, %ld pages, %ld bytes/page)\n", ((float
)npages * (float)pagesizeb) / 1000000.0, ((float)npages * (float)pagesizeb) / 1048576.0, npages,
pagesizeb);
```

After you have found the memory size, set the value of wr_throttle to 2% of the total size, as shown in the following table:

Memory Size	Value for wr_throttle
1 GByte	20 MBytes
4 GBytes	80 MBytes
16 GBytes	320 MBytes
64 GBytes	1.3 GBytes

```
:wr_throttle <eq> <value>
```

For eq, specify the Equipment Number for a file system.
For value, specify an integer number of kilobytes. If value=0, there is no limit.

File System Commands: Direct I/O Management

The commands in this section control I/O on Sun QFS file systems. They enable you to change the type of I/O for an individual file based on I/O size and history. If direct I/O is specified for a file, for example, through the `setfa(1)` command, these options are ignored and all I/O to regular files is direct, if possible.

These commands refer to both well-aligned and misaligned I/O.

- Well-aligned I/O occurs when the file offset falls on a 512-byte boundary and when the length of the I/O transfer is at least 512 bytes.
- Misaligned_ I/O occurs when the file offset does not fall on a 512-byte boundary and the length of the transfer is less than 512 bytes.

For more information about I/O and I/O management, see [Advanced Topics|#ahdfv].

The **dio_rd_form_min** and **dio_wr_form_min** Commands

These commands set the lower limits for well-aligned I/O to the specified number of 1024-byte blocks. Use the `dio_rd_form_min` command to set the number for read operations, and use the `dio_wr_form_min` command to set the number for write operations. By default,the minimum number of blocks is 256.

```
:dio_rd_form_min <eq> <value>
:dio_wr_form_min <eq> <value>
```

For eq, specify the Equipment Number for the file system.
For value, specify an integer number of 1024-byte blocks to use for the lower limit. If value=0, automatic I/O switching is disabled.

The **dio_rd_ill_min** and **dio_wr_ill_min** Commands

These commands set the lower limit for misaligned I/O to the specified number of 1024-byte blocks. Use the `dio_rd_ill_min` command to set the number for read operations, and use the `dio_wr_ill_min` command to set the number for write operations. By default,the minimum number of blocks is 256.

```
:dio_rd_ill_min <eq> <value>
:dio_wr_ill_min <eq> <value>
```

For value, specify an integer number of 1024-byte blocks to use for the lower limit. If value=0, automatic I/O switching is disabled.

The **dio_rd_consec** and **dio_wr_consec** Commands

These commands set the number of consecutive I/O transfers that can occur with a buffer size greater than the specified lower limits. By default, value=0, which means that no default direct read operations occur based on I/O sizes.

```
:dio_rd_consec <eq> <value>
:dio_wr_consec <eq> <value>
```

For eq, specify the Equipment Number for the file system.
For value, specify the number of consecutive I/O transfers with a buffer size greater than the specified lower limit. The specified lower limit is the value of `dio_rd_form_min` for aligned read operations or `dio_rd_ill_min` for misaligned read operations.
For more information, see one or more of the following commands or mount parameters:

- The `dio_rd_form_min` and `dio_wr_form_min` Commands
- The `dio_rd_ill_min` and `dio_wr_ill_min` Commands

The **dio_szero** and **nodio_szero** Commands

These commands set or clear the direct I/O sparse zeroing mount option.
The `dio_szero` command causes uninitialized areas of sparse files written with direct I/O to be zeroed when the area is accessed. This makes the sparse file behavior the same as that for paged I/O. By default, sparse files written by direct I/O do not have the uninitialized areas zeroed for performance reasons. The default is `nodio_szero`.

```
dio_szero <eq>
nodio_szero <eq>
```

For eq, specify the Equipment Number for the file system.

The **forcedirectio** and **noforcedirectio** Commands

These commands enable you to control whether direct I/O is used as the default I/O mode. By default, the I/O mode is buffered and uses the page cache. The `forcedirectio` command enables direct I/O for all transfers. The `noforcedirectio` command restores the default,buffered I/O.
When direct I/O is specified, the system transfers data directly between the users buffer and disk. Use direct I/O only if the file system is used for large, block-aligned, sequential I/O.

```
:forcedirectio <eq>
:noforcedirectio <eq>
```

For eq, specify the Equipment Number for the file system.
For more information about I/O, see [Advanced Topics|#ahdfv].

File System Commands: Sun QFS Shared File Systems

The following file system commands are supported on Sun QFS shared file systems only.

The **meta_timeo** Command

The `metatimeo` command sets the time limit for the Sun QFS shared file system metadata cache. The default value is 3. For more information about using this feature, see Retaining Cached Attributes: the `meta_timeo`=n Option.

```
:meta_timeo <eq> <interval>
```

For eq, specify the Equipment Number of the file system.
For interval, specify time in seconds. After this interval expires, the client host systems obtain a new copy of the metadata information from the

metadata server host.

The **mh_write** and **nomh_write** Commands

These commands enable or disable multihost read and write operations. For information about this feature, see Enabling Multiple Host Reads and Writes: the `mh_write` Option.

```
:mh_write <eq>
:nomh_write <eq>
```

For eq, specify the Equipment Number of the file system.

The **minallocsz** and **maxallocsz** Commands

These commands set the minimum and maximum block allocation size.

```
:minallocsz <eq> <value>
:maxallocsz <eq> <value>
```

For eq, specify the Equipment Number of the file system.
For value, and for more information about this feature, see Tuning Allocation Sizes: the `minallocsz=n` and `maxallocsz=n` Options.

The **rdlease**, **wrlease**, and **aplease** Commands

These commands control the amount of time granted for read, write, and append leases. The default time is 30 seconds. For information about this feature, see Using Leases in a Sun QFS Shared File System: the `rdlease=n`, `wrlease=n`, and `aplease=n` Options.

```
:rdlease <eq> <interval>
:wrlease <eq> <interval>
:aplease <eq> <interval>
```

For eq, specify the Equipment Number of the file system.
For interval, specify an integer number of seconds, `15`≤ interval≤ `600`. .

File System Commands: Miscellaneous

The following commands enable you to control leases, allocation sizes, and various other file system characteristics.

The **abr** and **noabr** Commands

These commands set or clear the application binary recovery (ABR) mount option.
For use in an Oracle RAC environment with Sun QFS asynchronous I/O (AIO) only. These mount options disable or enable ABR of software mirrors. They apply only to Sun QFS file systems built on Solaris Volume Manager mirrored volumes that support ABR.

```
:abr <eq>
:noabr <eq>
```

For eq, specify the Equipment Number for the file system.

The **dmr** and **nodmr** Commands

These commands set or clear the direct mirror reads (DMR) mount option.
For use in an Oracle Real Application Cluster (RAC) environment with Sun QFS AIO only. These mount options disable or enable DMR of software mirrors. They apply only to Sun QFS file systems built on Solaris Volume Manager mirrored volumes that support DMR.

```
:dmr <eq>
:nodmr <eq>
```

For eq, specify the Equipment Number for the file system.

The **invalid** interval Command

The invalid command specifies that the file system hold cached attributes for at least the specified number of seconds after a file is modified. You can specify this command only if the file system was mounted originally with the reader mount option. For information about mount options, see the mount_samfs(1M) man page.

```
:invalid <eq> <interval>
```

For eq, specify the Equipment Number for the file system.
For interval, specify the number of seconds to hold the attributes after file modification. For example, assume that interval=30. When you issue an ls(1) command, you might not see a newly created file appear in its output for 30 seconds after it has been created on its writer host.

The **mm_stripe** Command

The mm_stripe command sets the metadata stripe width for the file system to the specified number of 16-kilobyte disk allocation units (DAUs). The default is 1 DAU so that the file system writes one DAU of metadata to one LUN before switching to another LUN.

```
:mm_stripe <eq> <value>
```

For eq, specify the Equipment Number of the file system.
For value, specify either 0 or 1. If value=1, which is the default, the file system writes one DAU of metadata to one LUN before switching to another LUN. If value=0, the metadata is round-robined across all available metadata LUNs.

The **qwrite** and **noqwrite** Commands

The qwrite and noqwrite commands control the ability to perform simultaneous read and write operations to the same file from different threads. Specify qwrite only if file system users handle multiple simultaneous transactions to the same file. This is useful in database applications. The qwrite feature improves I/O performance by queuing multiple requests at the drive level. The qwrite specification is disabled for NFS reads or writes of the file system.

The default setting is noqwrite, so the file system disables simultaneous read and write operations to the same file. This is the mode defined by the UNIX vnode interface standard that gives exclusive access to only one writer and forces other writers and readers to wait.

```
:qwrite <eq>
:noqwrite <eq>
```

For eq, specify the Equipment Number of the file system.

The **refresh_at_eof** and **norefresh_at_eof** Commands

The refresh_at_eof and norefresh_at_eof commands can be used for fast updates to Sun QFS hosts that are mounted with the reader mount option in a multireader file system. This option ensures that the system refreshes the current file size when the read buffer exceeds the end of file. You can use this, for example, if the writer host system is appending to a file and the reader is issuing tail(1) commands with the -f option. The default is norefresh_at_eof.

```
:refresh_at_eof <eq>
:norefresh_at_eof <eq>
```

For eq, specify the Equipment Number of the file system.

The **suid** and **nosuid** Commands

The suid and nosuid commands control whether running programs are allowed to automatically change their owner IDs. For more information about the implications of using these mount options, see the suid and nosuid mount option descriptions on the mount_ufs (1M) man page and see the suid(2) man page.

```
:suid <eq>
:nosuid <eq>
```

For eq, specify the Equipment Number of the file system.

The **stripe** Command

The stripe command sets the stripe width for the file system to the specified number of disk allocation units (DAUs). The stripe width specifies that value multiplied by the DAU bytes are written to one LUN before switching to the next LUN. You can use the sammkfs(1M) -a command to set the DAU size on the file system when it is initialized.

```
:stripe <eq> <value>
```

For eq, specify the Equipment Number of the file system.
For value, specify an integer such that 0 < value < 255. If value=0, files are round-robin on each slice. The default value on file systems with an ms Equipment Type and on file systems with an ma Equipment Type with no striped group (gXXX) components is as follows:

- 128 kilobytes/DAU for DAUs < 128 kilobytes
- 1 for DAUs > 128 kilobytes

By default, value=0 on a Sun QFS shared file system.

By default, value=0 on file systems with an ma Equipment Type with any striped group (gXXX) components.

The system sets value=0 if mismatched striped groups exist.

For more information about file system types, see Design Basics and Configuring the File System.

The **sync_meta** Command

The sync_meta command determines whether metadata is written to disk every time it changes. If you are using this command on Sun QFS shared file system, also see Specifying the Frequency With Which Metadata Is Written: the sync_meta=n Option.

```
:sync_meta <eq> <value>
```

For eq, specify the Equipment Number of the file system.
For value, specify either 0 or 1, as follows:

- For value is 0, metadata is held in a buffer after it changes. For an unshared Sun QFS file system in which higher performance is desired, you can set value to 0. In this case, the system performs a delayed write operation in which metadata is held in a buffer before it is written to disk. This is the default for unshared file systems and for file systems that are not mounted as multireader file systems.
- For value is 1, metadata is written to disk every time it changes. This slows performance, but it increases data consistency. This is the default for Sun QFS file systems mounted as multireader file systems or as shared file systems. For a Sun QFS shared file system, value must be set to 1 if failover capability is required.

The **trace** and **notrace** Commands

The trace command enables tracing for a file system. The notrace command disables tracing. These are global directives that affect all operations. For more information about file system tracing, see the defaults.conf(4) man page.

```
:trace <eq>
:notrace <eq>
```

For eq, specify the Equipment Number of a file system.

Miscellaneous Commands

The following commands enable you to control tracing, open access to a disk device, and perform several other miscellaneous tasks.

The **clear** vsn Command

The `clear` command clears the specified VSN from the Removable Media Mount Requests display. For more information, see (p) - Removable Media Load Requests Display.

```
:clear <vsn>
:clear <vsn> <index>
```

For vsn, specify the volume to mount. Any process waiting for the VSN mount is aborted.
For index, specify the decimal ordinal of the VSN in the removable media display.

The **devlog** Command

The `devlog` command sets one or more events to be logged.

```
:devlog <eq>
:devlog <eq> <option>
```

For eq, specify the Equipment Number of a device.
For option, specify one or more event types. Possible event types are as follows: `all`, `date`, `default`, `detail`, `err`, `event`, `label`, `mig`, `module`, `msg`, none, `retry`, `stage`, `syserr`, and `time`. For information about these options, see the `defaults.conf`(4) man page.
If no option is specified, the system does not change the current events being logged for the eq specified.

The **diskvols** flag Command

The diskvols command sets or clears flags in the disk volume dictionary.

```
:diskvols <volume> +flag
:diskvols <volume> -flag
```

For volume, specify the volume in the disk volume dictionary.
For flag, specify one of the five flags in the `D samu`(1M) display. For information about the disk volume dictionary and the flags see (D) - Disk Volume Dictionary or the `samu`(1M) man page.

The **dtrace** Commands

The `dtrace` commands control the dtrace feature for one or more processes. The `dtrace` commands specify various tracing options.

```
:dtrace <daemon_name> on
:dtrace <daemon_name> off
:dtrace <daemon_name>.<variable> <value>
```

For daemon_name, specify the keyword `all` to affect all processes or a process name. If one of the following process names is specified, the tracing command affects that process only: `sam-archiverd`,`sam-catserverd`, `sam-fsd`, `sam-rftd`, `sam-recycler`, `sam-sharefsd`, and `sam-stagerd`.
For variable and value, specify one of the following variable and value pairs. The `defaults.conf`(4) man page contains comprehensive information about these arguments.

- `file` value
  Specify the name of a file to which trace files can be written. This can be a full path name.
- `options` value
  Specify a space-separated list of trace options.

- age value
  Specify the trace file rotation age. Note: Do not set an age of two minutes or less. If you do, the rotation never happens.
- size value
  Specify the size of the trace file at which rotation begins.

The **fs** Command

The fs command sets the file system to be displayed through the N display.

```
:fs <fsname>
```

For fsname, specify the name of the file system to be examined.

The **mount** Command

The mount command selects a Sun QFS file system.

```
:mount <mntpt>
```

For mntpt, specify the mount point of a file system.

The **open** Command

The open command enables access to the specified disk device. You must issue this command before you can use the read command, disk sector display (S), or file label display (F).

```
:open <eq>
```

For eq, specify the Equipment Number of a device.

The **read** Command

The read command reads the specified sector from the currently opened disk device. You must open the device before it can be read.

```
:read <addr>
```

For addr, specify the hexadecimal sector address.

The **refresh** Command

The refresh command sets the amount of time between samu(1M) screen refreshes.

```
:refresh <i>
```

For i, specify a time in seconds.

The **snap** Command

The snap command sends a snapshot of a display window to file. The default file is snapshots in the current working directory. To aid in problem reporting, take a snapshot of all the samu(1M) utility's displays. Each new snapshot is appended to the snapshots file. The file can be printed, examined using vi(1), or faxed to Sun Microsystems customer support staff.

```
:snap
:snap <filename>
```

For filename, specify the path of a file to receive the display information.

The ! shell_command Command

The ! command enables you to run a shell command without leaving the samu(1M) operator utility.

```
:! <shell_command>
```

For shell_command, specify a command.

End of Sun QFS File System Configuration and Administration Guide

About the Master Configuration File

Contents

About the Master Configuration File

The master configuration file /etc/opt/SUNWsamfs/mcf describes all devices that are under the control of, or used by, the Sun QFS or SAM-QFS software. When you create this ASCII file at system configuration time, you declare attributes for each device, and you group the devices in each file system into family sets.

The mcf file contains the information that these file systems need in order to identify and organize RAID and disk devices into file systems. It also contains entries for each automated library or device included in a file system. A sample mcf file is located in /opt/SUNWsamfs/examples/mcf.

Basic **mcf** File Structure

An mcf file consists of lines of specification code divided into six columns, or fields, as shown in the following example.

```
Equipment   Equipment  Equipment  Family  Device  Additional
Identifier  Number     Type       Set     State   Parameters
```

Follow these rules when entering data in the mcf file:

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

For more information about writing the mcf file, see the mcf(4) man page.

You can also use SAM-QFS Manager to automatically create an mcf file. For information about installing SAM-QFS Manager, see Installing SAM-QFS Manager. For information about using SAM-QFS Manager, see its online help.

mcf File Fields

- The Equipment Identifier Field (Required)
- The Equipment Number Field (Required)

- The Equipment Type Field (Required)
- The Family Set Field (Required)
- The Device State Field (Optional)
- The Additional Parameters Field (Optional)

The Equipment Identifier Field (Required)

The Equipment Identifier field identifies the physical file system device or removable media device. If this field contains the name of a file system, it is limited to 31 characters. For all other content, this field is limited to 127 characters.

Use the Equipment Identifier field to provide the following information:

Information	Identifer Length	Description
The file system name	31 characters	The file system name must be identical to the name in the Family Set field, and the subsequent lines in the `mcf` file must define all the disks or devices included in the file system. More than one file system can be declared in an `mcf` file. Typically, the first data line in an `mcf` file declares the first file system, and subsequent lines specify the devices included in the file system. Other file systems declared in the `mcf` file can be preceded by a blank comment line for readability. Note - File system names must start with an alphabetic character and can contain only alphabetic characters, numeric characters, or underscore (_) characters.
The `nodev` keyword	127 characters	The keyword `nodev` indicates that the system on which the `mcf` file resides is being used as a client host in a shared file system on a Solaris host. This keyword can appear in this field only as the Equipment Identifier for one or more metadata devices that reside on the metadata server. For more information about creating an `mcf` file for the members of a shared file system, see Configuring a Shared File System. Note - Do not use this keyword if your file system is in a Sun Cluster environment.
A disk partition or slice description	127 characters	A `/dev/` entry in this field identifies a disk partition or slice.
An automated library or optical drive description	127 characters	A `/dev/samst` entry identifies an automated library or optical drive. If you are configuring a network attached automated library, see Creating Parameters Files for Network Attached Automated Libraries.
A tape drive description	127 characters	This entry can be in one of two forms: • A `/dev/rmt` entry. • A path to a symbolic link that points to the same file to which the `/dev/rmt` link points. If you specify a tape drive in this manner, be sure to create the link before mounting the file system.

The Equipment Number Field (Required)

For each row in the `mcf` file, the Equipment Number (eq) field must contain a unique numeric identifier for the file system component or device being defined. This number must be an integer between 1 and 65534, inclusive.

> Tip -
> Use low numbers to keep the internal software tables small.

The Equipment Type Field (Required)

The required Equipment Type field provides information that the software uses to determine how to interact with a particular device. Enter the two- or three-character mnemonic for the device type.

Some equipment can use the generic equipment types of od (optical disk), tp (tape), and rb (robot). For a file system, the following table describes specific Equipment Type codes:

Table – Equipment Type Field

Equipment Type	Description
ms	Defines a file system that stores both data and metadata on the same device (an md device).
ma	Defines a file system that stores metadata on a separate device (an mm device). The data in an ma file system can be stored on md , mr, or g_XXX devices.
md	Defines a striped or round-robin device that uses dual allocation for storing file data. See also Dual and Single Allocation Schemes.
mm	Defines a metadata device for storing inode and other metadata information. You can specify multiple metadata devices. Metadata (including inodes, directories, allocation maps, and so on) on ma file systems is located on metadata devices, separated from the file data devices. By default, metadata is allocated using round-robin allocation if you have multiple metadata devices.
mr	Defines a round-robin or striped data device that uses single allocation for storing file data. See also Dual and Single Allocation Schemes.
gXXX	Defines a striped data device. Striped groups start with the letter g followed by a number. The number must be an integer between 0 and 127, inclusive; for example, g12.All members in a striped group must be the same type and size. Different striped groups within one file system are not required to have the same number of members. md, mr, and gXXX devices cannot be mixed in one file system. Data can be striped (if all groups contain the same number of devices) or round-robin between groups. The default is round-robin.

Besides the file system equipment types, other codes are used to identify automated libraries and other devices. For more information about specific equipment types, see the mcf(4) man page.

The Family Set Field (Required)

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

Family set names must start with an alphabetic character and can contain only alphabetic characters, numeric characters, or underscore (_) characters. The family set name cannot be longer than 31 characters.

The family set name can be one of the following:

Family Set	Description
File system name	All disk devices in the file system must use the same file system name in this field. The software uses the family set name to group devices together as a file system. It physically records the family set name on all of the devices in the file system when the sammkfs(1M) command is issued. You can change this name by using the -F and -R options together in the samfsck(1M) command. For more information about the sammkfs(1M) command, see the sammkfs(1M) man page. For more information about the samfsck(1M) command, see the samfsck(1M) man page.
Automated library identifier	The library and all its associated drive devices must use the same identifier.
-	The dash character (-) indicates a standalone removable media device.

You can create a comment that is associated with a specific family set by inserting the identifier #family-set-name: just before the first device in that family set. Any comments that are added between that comment line and the last device in the family set will be associated with that family set. If the family set is later deleted through the SAM-QFS Manager software, any related comments will also be deleted from the mcf file.

The Device State Field (Optional)

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

The Additional Parameters Field (Optional)

For an automated library device, the Additional Parameters field is optional and can be left blank. By default, library catalog files are written to `/var/opt/SUNWsamfs/catalog/`family-set-name. Use this field if you want to specify an alternative path to the library catalog file.

For a shared file system, this field must contain the keyword `shared`.

For other entries, enter a dash (-) or leave this field blank.

Examples of mcf Files

Each file system configuration is unique. System requirements and actual hardware differ from site to site. The following code examples show sample `mcf` files. More complete examples that include information on how you can duplicate the configuration are available in mcf File Examples.

Example – Example **mcf** File Showing Striped Groups

This example shows an `mcf` file for a Sun QFS file system with two striped groups.

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

Example – Example **mcf** File Showing Three File Systems

This example shows an `mcf` file with three file systems.

```
# SAM-QFS file system configuration example
#
# Equipment  Eq  Eq   Fam. Dev. Additional
# Identifier Ord Type Set State Parameters
#----------- --- -- ------ ------ -----------------
qfs1 10 ma qfs1 -
/dev/dsk/c1t13d0s6 11 mm qfs1 -
/dev/dsk/c1t12d0s6 12 mr qfs1 -
#
qfs2 20 ma qfs2 -
/dev/dsk/c1t5d0s6 21 mm qfs2 -
/dev/dsk/c5t1d0s6 22 mr qfs2 -
#
qfs3 30 ma qfs3 -
/dev/dsk/c7t1d0s3 31 mm qfs3 -
/dev/dsk/c6t1d0s6 32 mr qfs3 -
/dev/dsk/c6t1d0s3 33 mr qfs3 -
/dev/dsk/c5t1d0s3 34 mr qfs3 -
```

Example – Example **mcf** File Showing a File System and a Library

This example shows an `mcf` file with one archiving file system that uses `md` devices. This `mcf` file also defines a tape library.

```
# Equipment Eq Eq Fam. Dev. Additional
# Identifier Ord Type Set State Parameters
#----------- --- -- ------ ------ ----------
samfs1 10 ma samfs1 -
/dev/dsk/c1t2d0s6 11 mm samfs1 -
/dev/dsk/c1t3d0s6 12 md samfs1 -
/dev/dsk/c1t4d0s6 13 md samfs1 -
/dev/dsk/c1t5d0s6 14 md samfs1 -
# scalar 1000 and 12 AIT tape drives
/dev/samst/c5t0u0 30 rb robot1 -
/dev/rmt/4cbn 101 tp robot1 on
/dev/rmt/5cbn 102 tp robot1 on
/dev/rmt/6cbn 103 tp robot1 on
/dev/rmt/7cbn 104 tp robot1 off
/dev/rmt/10cbn 105 tp robot1 on
/dev/rmt/11cbn 106 tp robot1 on
/dev/rmt/3cbn 107 tp robot1 on
/dev/rmt/2cbn 108 tp robot1 on
/dev/rmt/1cbn 109 tp robot1 on
/dev/rmt/0cbn 110 tp robot1 on
/dev/rmt/9cbn 111 tp robot1 on
/dev/rmt/8cbn 112 tp robot1 on
```

Interactions Among File Settings, Options, and Directives

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

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

For information about the various ways to specify mount options, see Setting Up Mount Parameters.

Related Topics

- How to Create the `mcf` File Manually
- How to Create the `mcf` File Using SAM-QFS Manager
- mcf File Examples

mcf File Examples

Contents

---

mcf File Examples

The master configuration file, /etc/opt/SUNWsamfs/mcf, defines the topology of the equipment managed by the file system. This file specifies the devices and file systems included in the environment and contains information that enables you to identify the disk slices to be used and to organize them into file systems.

This section provides some specific examples of mcf files for various types of file systems.

Configuration Examples for Local File Systems

Use the configuration examples in this section for configuring the mcf file for a file system to be installed on a single Solaris OS host.

For mcf examples that you can use in a Sun Cluster environment, see Configuration Examples for Highly Available File Systems.

Simple File System Configuration Example

This example shows how to configure two file systems using a server that has a SCSI-attached Sun StorageTek Multipack desktop array.

You can use the format(1M) command to determine how the disks are partitioned. The following example shows the format(1M) command output.

> Note -
> Only the last lines of format(1M) output are shown.

Example – ***format***(1M) Command Output for Configuration Example 1

```
# format < /dev/null
Searching for disks...done

AVAILABLE DISK SELECTIONS:
                0. c0t10d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
                    /sbus@3,0/SUNW,fas@3,8800000/sd@a,0
                1. c0t11d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
                    /sbus@3,0/SUNW,fas@3,8800000/sd@b,0
                2. c6t2d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
                    /pci@7,4000/SUNW,isptwo@3/sd@2,0
                3. c6t3d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
                    /pci@7,4000/SUNW,isptwo@3/sd@3,0
                4. c6t4d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
                        /pci@7,4000/SUNW,isptwo@3/sd@4,0
                5. c6t5d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
                    /pci@7,4000/SUNW,isptwo@3/sd@5,0
                6. c8t2d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
                    /pci@b,4000/SUNW,isptwo@3/sd@2,0
                7. c8t3d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
                    /pci@b,4000/SUNW,isptwo@3/sd@3,0
                8. c8t4d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
                    /pci@b,4000/SUNW,isptwo@3/sd@4,0
                9. c8t5d0 <SUN9.0G cyl 4924 alt 2 hd 27 sec 133>
                    /pci@b,4000/SUNW,isptwo@3/sd@5,0
Specify disk (enter its number):

# format /dev/rdsk/c6t2d0s2
.
.
.
Part Tag Flag Cylinders Size Blocks
        0 unassigned wm 0 0 (0/0/0) 0
        1 unassigned wm 0 0 (0/0/0) 0
        2 backup wu 0 - 4923 8.43GB (4924/0/0) 17682084
        3 unassigned wm 0 0 (0/0/0) 0
        4 unassigned wm 0 - 1229 2.11GB (1230/0/0) 4416930
        5 unassigned wm 1230 - 2459 2.11GB (1230/0/0) 4416930
        6 unassigned wm 2460 - 3689 2.11GB (1230/0/0) 4416930
        7 unassigned wm 3690 - 4919 2.11GB (1230/0/0) 4416930
```

How to Configure the System for Simple Example

Begin writing the `mcf` file for this configuration example by defining the file system and its disk partitions, as follows:

1. Write the `mcf` file.
2. Make an `ma` entry for the first file system (`qfs1`).
3. Using the information from the output of the `format` command, make an `mm` entry listing the partitions that constitute the metadata for the `qfs1` file system.
4. Using the information from the output of the `format` command, make a series of `mr` entries listing the partitions that constitute the file data for the `qfs1` file system.
5. Make similar entries for the second file system (`qfs2`).
   The finished `mcf` file defines the following two file systems:
   - The `qfs1` file system, which is created on slice 4 of the following disks: `c8t2d0` (metadata), `c6t2d0` (file data), and `c6t3d0` (file data).
   - The `qfs2` file system, which is created on slice 5 of the following disks: `c8t2d0` (metadata), `c6t2d0` (file data), and `c6t3d0` (file data).
     The following code example shows the resulting `mcf` file.

```
# cat /etc/opt/SUNWsamfs/mcf
#
# Equipment         Eq   Eq     Family   Device   Additional
# Identifier        Ord  Type    Set     State    Parameters
#-----------        ---  ----   ------   ------   ----------
#
*qfs1 10 ma qfs1 on*
/dev/dsk/c8t2d0s4    11    mm   qfs1       on
/dev/dsk/c6t2d0s4    12    mr   qfs1       on
/dev/dsk/c6t3d0s4    13    mr   qfs1       on
#
*qfs2 20 ma qfs2 on*
*/dev/dsk/c8t2d0s5 21 mm qfs2 on*
*/dev/dsk/c6t2d0s5 22 mr qfs2 on*
*/dev/dsk/c6t3d0s5 23 mr qfs2 on*
```

6. Modify the `/etc/vfstab` file.
   Make entries in the `/etc/vfstab` file for the `qfs1` and `qfs2` file systems that you defined in the `mcf` file. The last two lines in the code example below show entries for these new file systems.
   For a description of the fields in the `/etc/vfstab` file, see Table 3-2.

```
# cat /etc/vfstab
# device            device                    file            mount
# to                to                  mount  system   fsck   at    mount
# mount             fsck                point  type     pass   boot  params
# -----             ----                -----  ----     ----   ----  ------
fd                  -                   /dev/fd  fd       -     no    -
/proc               -                   /proc    proc     -     no    -
/dev/dsk/c0t10d0s1  -                   -        swap     -     no    -
/dev/dsk/c0t10d0s0  /dev/rdsk/c0t10d0s0  /        ufs      1     no    logging
swap                -                   /tmp     tmpfs    -     yes   -
*qfs1 - /qfs1 samfs - yes stripe=1*
*qfs2 - /qfs2 samfs - yes stripe=1*
```

Round-Robin Configuration Example

This example illustrates the configuration of a file system (called `qfs3`) that uses round-robin allocation on four disk drives.

This example assumes the following:

- The metadata device is a single partition (`s1`) used on controller 8, disk 4.
- The data devices consist of four disks attached to controller 6. Each disk is on a separate target (1–4).

How to Configure the System for Round-Robin Allocation

This example introduces the round-robin data layout. For more information about data layout, see the Sun StorageTek QFS File System Configuration and Administration Guide.

1. Write the mcf file as described in Configuration Example 1.
   The following code example shows the mcf file for this round-robin disk configuration.

```
# cat /etc/opt/SUNWsamfs/mcf
#
# Equipment         Eq    Eq     Family    Device    Additional
# Identifier        Ord   Type    Set      State     Parameters
#-----------        ---   ----   ------    ------    ----------
#
*qfs3 10 ma qfs3 on*
/dev/dsk/c8t4d0s4    11    mm    qfs3       on
/dev/dsk/c6t2d0s4    12    mr    qfs3       on
/dev/dsk/c6t3d0s4    13    mr    qfs3       on
/dev/dsk/c6t4d0s4    14    mr    qfs3       on
/dev/dsk/c6t5d0s4    15    mr    qfs3       on
```

2. Modify the /etc/vfstab file.
   Edit the /etc/vfstab file to explicitly set round-robin allocation on the file system by specifying stripe=0 in the mount params field. The following code example shows stripe=0 for the qfs3 file system.
   For a description of the fields in the /etc/vfstab file, see Table 3-2.

```
# cat /etc/vfstab
#device              device                      file           mount
#to                  to                   mount  system  fsck   at    mount
#mount               fsck                 point  type    pass   boot  params
#-----               ----                 -----  ----    ----   ----  ------
fd                   -                    /dev/fd  fd     -      no    -
/proc                -                    /proc  proc    -      no    -
/dev/dsk/c0t10d0s1   -                    -      swap    -      no    -
/dev/dsk/c0t10d0s0   /dev/rdsk/c0t10d0s0  /      ufs     1      no    logging
swap                 -                    /tmp   tmpfs   -      yes   -
*qfs3 - /qfs3 samfs - yes stripe=0*
```

3. Initialize the file system by using the sammkfs(1M) command.
   The default disk allocation unit (DAU) is 64 kilobytes, but the following example sets the DAU size to 128 kilobytes:

```
# sammkfs -a 128 qfs3
```

Local Striping Configuration Example

This example illustrates the configuration of a file system (called qfs4) that stripes file data to four disk drives. This example assumes the following:

- The metadata device is a single partition (s6) used on controller 0, logical unit number (LUN) 0.
- The data devices consist of four disks attached to controller 6. Each disk is on a separate target (2–5).

How to Configure the System for Local Striping

1. Write the mcf file as shown in Configuration Example 1.
   The following code example shows the mcf file for this striped disk configuration.

```
# Equipment         Eq   Eq     Family   Device   Additional
# Identifier        Ord  Type    Set     State    Parameters
#-----------        ---  ----   ------   ------   ----------
#
*qfs4 40 ma qfs4 on*
/dev/dsk/c8t4d0s4    41    mm   qfs4       on
/dev/dsk/c6t2d0s4    42    mr   qfs4       on
/dev/dsk/c6t3d0s4    43    mr   qfs4       on
/dev/dsk/c6t4d0s4    44    mr   qfs4       on
/dev/dsk/c6t5d0s4    45    mr   qfs4       on
```

2. Modify the `/etc/vfstab` file.
   Set the stripe width by using the `stripe=` option. The following code example shows the `/etc/vfstab` file with a mount parameter of `stripe=1` set for the `qfs4` file system.
   For a description of the fields in the `/etc/vfstab` file, see Table 3-2.

```
# cat /etc/vfstab
#
#device             device                        file         mount
#to                 to                   mount    system fsck  at    mount
#mount              fsck                 point    type   pass  boot  params
#-----              ----                 -----    -----  ----  ----  ------
fd                  -                    /dev/fd  fd     -     no    -
/proc               -                    /proc    proc   -     no    -
/dev/dsk/c0t10d0s1  -                    -        swap   -     no    -
/dev/dsk/c0t10d0s0  /dev/rdsk/c0t10d0s0  /        ufs    1     no    logging
swap                -                    /tmp     tmpfs  -     yes   -
*qfs4 - /qfs4 samfs - yes stripe=1*
```

   The `stripe=1` specification stripes file data across all four of the `mr` data disks with a stripe width of one DAU. The DAU is the allocation unit you set when you use the `sammkfs`(1M) command to initialize the file system.

3. Initialize the Sun StorageTek QFS file system by using the `sammkfs`(1M) command.
   The following example sets the DAU size to 128 kilobytes:

```
# sammkfs -a 128 qfs4
```

   With this striped disk configuration, any file written to this file system is striped across all of the devices in increments of 128 kilobytes. Files less than the aggregate stripe width times the number of devices still use 128 kilobytes of disk space. Files larger than 128 kilobytes have space allocated for them as needed in total space increments of 128 kilobytes.

Striped Group Configuration Example

Striped groups enable you to build RAID-0 devices of separate disk devices. With striped groups, however, there is only one DAU per striped group. This method of writing large, effective DAUs across RAID devices saves system update time and supports high-speed sequential I/O. Striped groups are useful for writing very large files to groups of disk devices.

> Note -
> A DAU is the minimum disk space allocated. The minimum disk space allocated in a striped group is as follows:
> allocation-unit x number of disks in the group
> Writing a single byte of data consumes a DAU on every member of the striped group. Make sure that you understand the effects of using striped groups with your file system.

The devices within a striped group must be the same size. It is not possible to increase the size of a striped group. You can add additional striped groups to the file system, however.

This example illustrates the configuration of a file system (called `qfs5`) that separates the metadata onto a low-latency disk. The `mcf` file defines two striped groups on four drives. This example assumes the following:

- The metadata device is a single partition (s5) used on controller 8, disk 4.
- The data devices consist of four disks (two groups of two identical disks) attached to controller 6. Each disk is on a separate target (2–5).

How to Configure the System for Striped Groups

1. Write the `mcf` file as shown in Configuration Example 1.
   The following code example shows a sample `mcf` file for a striped group configuration.

```
# cat /etc/opt/SUNWsamfs/mcf
#
# Equipment         Eq   Eq     Family   Device   Additional
# Identifier        Ord  Type    Set     State    Parameters
#-----------        ---  ----   ------   ------   ----------
#
*qfs5 50 ma qfs5 on*
/dev/dsk/c8t4d0s5    51    mm   qfs5       on
/dev/dsk/c6t2d0s5    52    g0   qfs5       on
/dev/dsk/c6t3d0s5    53    g0   qfs5       on
/dev/dsk/c6t4d0s5    54    g1   qfs5       on
/dev/dsk/c6t5d0s5    55    g1   qfs5       on
```

2. Modify the `/etc/vfstab` file.
   Set the stripe width by using the `stripe=` option. The following code example shows the `/etc/vfstab` file with a mount parameter of `stripe=0`, which specifies round-robin allocation between striped group `g0` and striped group `g1`.
   For a description of the fields in the `/etc/vfstab` file, see “Fields in the /etc/vfstab File” on page 35.

```
# cat /etc/vfstab
#device             device                       file          mount
#to                 to                  mount    system  fsck  at    mount
#mount              fsck                point    type    pass  boot  params
#-----              ----                -----    ----    ----  ----  ------
fd                  -                   /dev/fd  fd      -     no    -
/proc               -                   /proc    proc    -     no    -
/dev/dsk/c0t10d0s1  -                   -        swap    -     no    -
/dev/dsk/c0t10d0s0  /dev/rdsk/c0t10d0s0 /        ufs     1     no    logging
swap                -                   /tmp     tmpfs   -     yes   -
*qfs5 - /qfs5 samfs - yes stripe=0*
```

3. Initialize the file system by using the `sammkfs`(1M) command.
   The{{– a}} option is not used with striped groups because the DAU is equal to the size of an allocation, or the size, of each group.

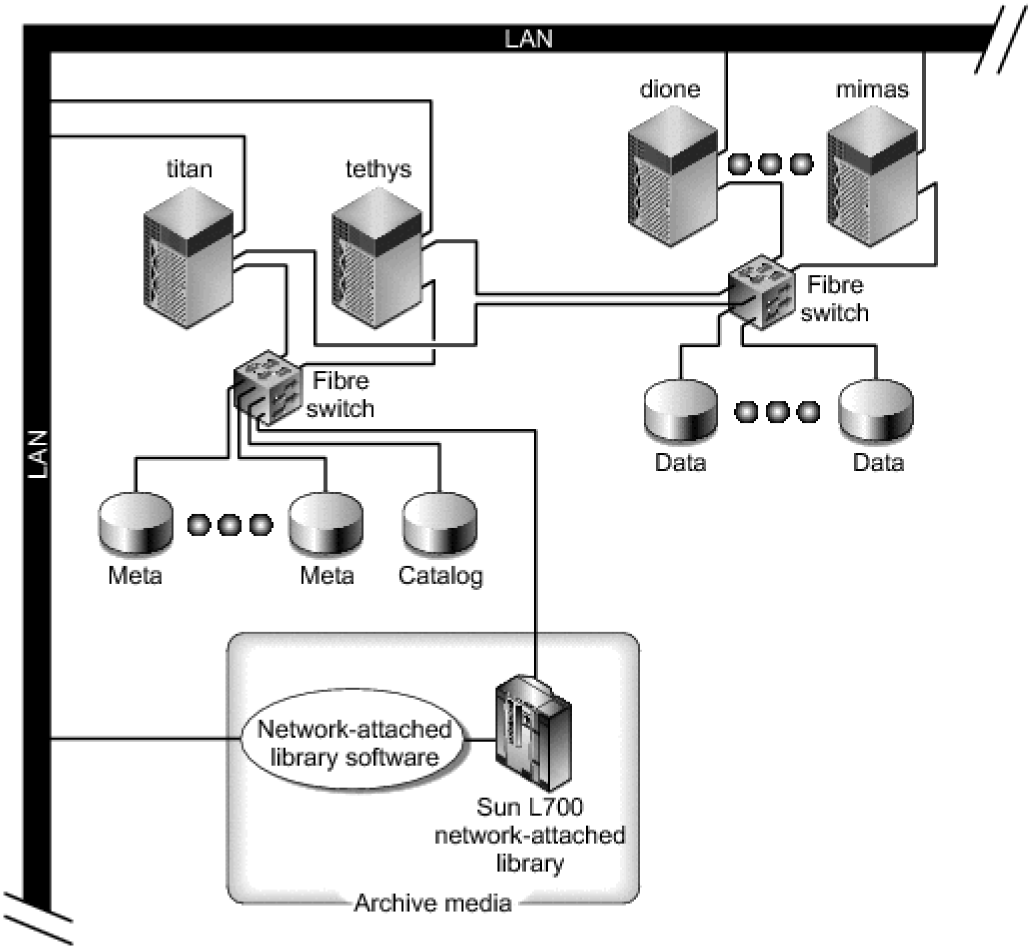
```
# sammkfs qfs5
```

   In this example, there are two striped groups, `g0` and `g1`. With `stripe=0` in `/etc/vfstab`, devices 12 and 13 are striped; devices 14 and 15 are striped; and files are round-robined around the two striped groups. A striped group is treated as a bound entity. After you configure a stripe group, you cannot change it without issuing another `sammkfs`(1M) command.

Configuration Example for a Shared File System on a Solaris OS Platform

This figure illustrates a shared file system configuration in an archiving environment.

Figure – Shared File System Configuration



The above figure shows four network attached hosts: `titan`, `tethys`, `dione`, and `mimas`. The `tethys`, `dione`, and `mimas` hosts are clients, and `titan` is the current metadata server. The `tethys` host is a potential metadata server.

The archive media consist of a network attached library and tape drives that are fibre-attached to `titan` and `tethys`. In addition, the archive media catalog resides in a file system that is mounted on the current metadata server, `titan`.

Metadata travels to and from the clients to the metadata server over the network. The metadata server makes all modifications to the name space, thereby keeping the metadata consistent. The metadata server also provides the locking capability, the block allocation, and the block deallocation.

Several metadata disks are connected to `titan` and `tethys` and can be accessed only by the potential metadata servers. If `titan` were unavailable, you could change the metadata server to `tethys`, and the library, tape drives, and catalog could be accessed by `tethys` as part of the Sun StorageTek QFS shared file system. The data disks are connected to all four hosts by a Fibre Channel (FC) connection.

How to Configure the Shared File System

1. Issue the `format`(1M) command and examine its output.
   Make sure that the metadata disk partitions configured for the shared file system mount point are connected to the potential metadata servers. Also make sure that the data disk partitions configured for the shared file system are connected to the potential metadata servers and to all the client hosts in this file system.
   If your host supports multipath I/O drivers, individual devices shown in the output of the `format`(1M) command might display multiple controllers. These correspond to the multiple paths to the actual devices.
   The following code example shows the `format`(1M) command output for `titan`. There is one metadata disk on controller 2, and there are three data disks on controller 3.

```
titan<28>format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c1t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w2100002037e9c296,0
       1. c2t2100002037E2C5DAd0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w2100002037e9c296,0
       2. c2t50020F23000065EEd0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w50020f23000065ee,0
       3. c3t50020F2300005D22d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@8,600000/SUNW,qlc@1/fp@0,0/ssd@w50020f2300005d22,0
       4. c3t50020F2300006099d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@8,600000/SUNW,qlc@1/fp@0,0/ssd@w50020f2300006099,0
       5. c3t50020F230000651Cd0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@8,600000/SUNW,qlc@1/fp@0,0/ssd@w50020f230000651c,0
```

The following code example shows the `format`(1M) command output for `tethys`. There is one metadata disk on controller 2, and there are four data disks on controller 7.

```
tethys<1>format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c0t1d0 <IBM-DNES-318350Y-SA60 cyl 11112 alt 2 hd 10 sec 320>
          /pci@1f,4000/scsi@3/sd@1,0
       1. c2t2100002037E9C296d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /pci@8,600000/SUNW,qlc@4/fp@0,0/ssd@w2100002037e9c296,0
       2. c2t50020F23000065EEd0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@4/ssd@w50020f23000065ee,0
       3. c7t50020F2300005D22d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@5/ssd@w50020f2300005d22,0
       4. c7t50020F2300006099d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@5/ssd@w50020f2300006099,0
       5. c7t50020F230000651Cd0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
          /pci@1f,4000/SUNW,qlc@5/ssd@w50020f230000651c,0
```

Note the following in above code example:

- The data disks on `titan`'s controller 3 are the same disks as on `tethys`'s controller 7. You can verify this by looking at the World Wide Name, which is the last component in the device name. For `titan`'s number 3 disk, the World Wide Name is `50020f2300005d22`. This is the same name as number 3 on controller 7 on `tethys`.
- For `titan`'s metadata disk, the World Wide Name is `50020F23000065EE`. This is the same metadata disk as `tethys`'s controller 2, target 0.
  The following code example shows the `format`(1M) command output for `mimas`. This shows three data disks on controller 1 and no metadata disks.

  ```
  mimas&lt;9>format
  Searching for disks...done


  AVAILABLE DISK SELECTIONS:
         0. c0t0d0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
            /pci@1f,4000/scsi@3/sd@0,0
         1. c1t50020F2300005D22d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
            /pci@1f,4000/SUNW,qlc@4/fp@0,0/ssd@w50020f2300005d22,0
         2. c1t50020F2300006099d0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
            /pci@1f,4000/SUNW,qlc@4/fp@0,0/ssd@w50020f2300006099,0
         3. c1t50020F230000651Cd0 <SUN-T300-0116 cyl 34901 alt 2 hd 128 sec 256>
            /pci@1f,4000/SUNW,qlc@4/fp@0,0/ssd@w50020f230000651c,0
  ```

As shown in the above code examples, the data disks on `titan`'s controller 3 are the same disks as those on `mimas`'s controller 1. You can verify this by looking at the World Wide Name, which is the last component in the device name.

> **Note -**
> All the data disk partitions must be connected and accessible from all the hosts that share this file system. All the disk partitions, for both data and metadata, must be connected and accessible to all potential metadata servers. You can use the `format`(1M) command to verify these connections.
> For some storage devices, it is possible that the `format`(1M) command's output does not present unique worldwide Names. If you find that this is the case, see the `libdevid`(3LIB) man page for information about finding such devices on different hosts.

2. Use `vi`(1) or another editor to create the `mcf` file on the metadata server.
   The only difference between the `mcf` file of a shared file system and an unshared file system is the presence of the `shared` keyword in the Additional Parameters field of the file system name line for a shared file system.

   > **Note -**
   > If file systems are already operational on the shared file system's metadata server or on any of the client host systems, select a Family Set name and select equipment numbers that do not conflict with existing Family Set names or equipment numbers on any host that will be included in the shared file system.

   The following code example shows a fragment of the `mcf` file for `titan` that defines several disks for use in the shared file system. It shows the `shared` keyword in the Additional Parameters field on the file system name line.

```
# Equipment                        Eq   Eq    Family   Dev   Addl
# Identifier                       Ord  Type  Set      Stat  Params
------------                       ---  ----  ------   ----  ------
sharefs1                           10   ma    sharefs1 on    shared
/dev/dsk/c2t50020F23000065EEd0s6 11   mm    sharefs1 on
/dev/dsk/c3t50020F2300005D22d0s6 12   mr    sharefs1 on
/dev/dsk/c3t50020F2300006099d0s6 13   mr    sharefs1 on
/dev/dsk/c3t50020F230000651Cd0s6 14   mr    sharefs1 on
```

Configuration Examples for Highly Available File Systems

The Sun Cluster software moves a highly available file system from a failing node to a viable node in the event of a node failure.

Each node in the Sun Cluster environment that can host this file system must have an `mcf` file. During the file system configuration process, you copy `mcf` file lines from the metadata server's `mcf` file to other nodes in the Sun Cluster environment. For more information, see Editing `mcf` Files on Other Hosts.

How to Create an mcf File for a Highly Available File System

The procedure for creating an `mcf` file for a highly available file system is as follows:

1. Make an `ma` entry for the file system.
2. Make an `mm` entry listing the partitions that constitute the metadata for the `qfs1` file system.
3. Make a series of `mr`, `gXXX`, or `md` entries listing the partitions that constitute the file data for the `qfs1` file system.
   You can use the `scdidadm`(1M) command to determine the partitions to use.
   Example 1. The following code example shows an `mcf` file entry for a highly available file system that uses raw devices.

```
#Equipment            Eq   Eq     Family   Additional
#Identifier           Ord  Type   Set      Parameters
#-------------------- ---  ----   ------   ----------
qfs1                   1   ma    qfs1     on
/dev/global/dsk/d4s0  11   mm    qfs1
/dev/global/dsk/d5s0  12   mr    qfs1
/dev/global/dsk/d6s0  13   mr    qfs1
/dev/global/dsk/d7s0  14   mr    qfs1
```

Example 2. The following code example shows an `mcf` file entry for a highly available file system that uses Solaris Volume Manager metadevices. In this example, the Solaris Volume Manager metaset in use is named `red`.

```
#Equipment            Eq   Eq     Family   Additional
#Identifier           Ord  Type   Set      Parameters
#-------------------- ---  ----   ------   ----------
qfs1                   1   ma    qfs1     on
/dev/md/red/dsk/d0s0  11   mm    qfs1
/dev/md/red/dsk/d1s0  12   mr    qfs1
```

Example 3. The following code example shows an `mcf` file entry for a highly available file system that uses VxVm devices.

```
#Equipment            Eq   Eq     Family   Additional
#Identifier           Ord  Type   Set      Parameters
#-------------------- ---  ----   ------   ----------
qfs1                   1   ma    qfs1     on
/dev/vx/dsk/oradg/m1  11   mm    qfs1
/dev/vx/dsk/oradg/m2  12   mr    qfs1
```

Configuration Example for a Shared File System on a Sun Cluster Platform

In this example, `ash` and `elm` are nodes in a Sun Cluster environment. Host `ash` is the metadata server. The keyword `shared` in this example's `mcf` file indicates to the system that this is a shared file system. This example builds upon Example – Verifying Devices and Device Redundancy.

How to Create an mcf File for a Shared File System in a Sun Cluster Environment

You must create the `mcf` file on the node that you want to designate as the metadata server. The procedure for creating an `mcf` file for a shared file system in a Sun Cluster environment is as follows:

1. Use the `scdidadm`(1M) `-L` command to obtain information about the devices included in the Sun Cluster environment.
   The `scdidadm`(1M) command administers the device identifier (DID) devices. The `-L` option lists all the DID device paths, including those on all nodes in the Sun Cluster environment.
   The following code example uses Sun StorageTek T3 arrays in a RAID-5 configuration. The output shows that you can use devices 4 through 9 for configuring the disk cache for a shared file system.

```
ash# scdidadm -L
1       ash:/dev/rdsk/c0t6d0        /dev/did/rdsk/d1
2       ash:/dev/rdsk/c1t1d0        /dev/did/rdsk/d2
3       ash:/dev/rdsk/c1t0d0        /dev/did/rdsk/d3
4       elm:/dev/rdsk/c6t50020F2300004921d1 /dev/did/rdsk/d4
4       ash:/dev/rdsk/c5t50020F2300004921d1 /dev/did/rdsk/d4
5       elm:/dev/rdsk/c6t50020F2300004921d0 /dev/did/rdsk/d5
5       ash:/dev/rdsk/c5t50020F2300004921d0 /dev/did/rdsk/d5
6       elm:/dev/rdsk/c6t50020F23000049CBd1 /dev/did/rdsk/d6
6       ash:/dev/rdsk/c5t50020F23000049CBd1 /dev/did/rdsk/d6
7       elm:/dev/rdsk/c6t50020F23000049CBd0 /dev/did/rdsk/d7
7       ash:/dev/rdsk/c5t50020F23000049CBd0 /dev/did/rdsk/d7
8       elm:/dev/rdsk/c6t50020F23000055A8d0 /dev/did/rdsk/d8
8       ash:/dev/rdsk/c5t50020F23000055A8d0 /dev/did/rdsk/d8
9       elm:/dev/rdsk/c6t50020F23000078F1d0 /dev/did/rdsk/d9
9       ash:/dev/rdsk/c5t50020F23000078F1d0 /dev/did/rdsk/d9
10      elm:/dev/rdsk/c0t6d0        /dev/did/rdsk/d10
11      elm:/dev/rdsk/c1t1d0        /dev/did/rdsk/d11
12      elm:/dev/rdsk/c1t0d0        /dev/did/rdsk/d12
```

2. Using the output from the `scdidadm`(1M) `-L` command, use the `format`(1M) command to display the information for the devices in the Sun Cluster environment. The following code example shows the `format` command output from all the `/dev/did` devices. You will need this information when you build the `mcf` file.

```
ash# format /dev/did/rdsk/d4s2
selecting /dev/did/rdsk/d4s2

Primary label contents:

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 64 sec 32>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =   64
nsect       =   32
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264       16.86GB    (17265/0/0) 35358720
  1        usr    wm   17265 - 34529       16.86GB    (17265/0/0) 35358720
  2     backup    wu       0 - 34529       33.72GB    (34530/0/0) 70717440
  3 unassigned    wu       0                0         (0/0/0)            0
  4 unassigned    wu       0                0         (0/0/0)            0
  5 unassigned    wu       0                0         (0/0/0)            0
  6 unassigned    wu       0                0         (0/0/0)            0
  7 unassigned    wu       0                0         (0/0/0)            0

ash# format /dev/did/rdsk/d5s2
selecting /dev/did/rdsk/d5s2

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 192 sec 64>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =  192
nsect       =   64
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264      101.16GB    (17265/0/0) 212152320
  1        usr    wm   17265 - 34529      101.16GB    (17265/0/0) 212152320
  2     backup    wu       0 - 34529      202.32GB    (34530/0/0) 424304640
  3 unassigned    wu       0                0         (0/0/0)             0
  4 unassigned    wu       0                0         (0/0/0)             0
  5 unassigned    wu       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7 unassigned    wu       0                0         (0/0/0)             0

ash# format /dev/did/rdsk/d6s2
selecting /dev/did/rdsk/d6s2
```

```
Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 64 sec 32>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =   64
nsect       =   32
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264       16.86GB    (17265/0/0) 35358720
  1        usr    wm   17265 - 34529       16.86GB    (17265/0/0) 35358720
  2     backup    wu       0 - 34529       33.72GB    (34530/0/0) 70717440
  3 unassigned    wu       0                0         (0/0/0)            0
  4 unassigned    wu       0                0         (0/0/0)            0
  5 unassigned    wu       0                0         (0/0/0)            0
  6 unassigned    wu       0                0         (0/0/0)            0
  7 unassigned    wu       0                0         (0/0/0)            0


ash# format /dev/did/rdsk/d7s2
selecting /dev/did/rdsk/d7s2

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 192 sec 64>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =  192
nsect       =   64
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264      101.16GB    (17265/0/0) 212152320
  1        usr    wm   17265 - 34529      101.16GB    (17265/0/0) 212152320
  2     backup    wu       0 - 34529      202.32GB    (34530/0/0) 424304640
  3 unassigned    wu       0                0         (0/0/0)             0
  4 unassigned    wu       0                0         (0/0/0)             0
  5 unassigned    wu       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7 unassigned    wu       0                0         (0/0/0)             0

ash# format /dev/did/rdsk/d8s2
selecting /dev/did/rdsk/d8s2

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 128 sec 128>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =  128
nsect       =  128
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264      134.88GB    (17265/0/0) 282869760
  1        usr    wm   17265 - 34529      134.88GB    (17265/0/0) 282869760
  2     backup    wm       0 - 34529      269.77GB    (34530/0/0) 565739520
  3 unassigned    wu       0                0         (0/0/0)             0
  4 unassigned    wu       0                0         (0/0/0)             0
  5 unassigned    wu       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7 unassigned    wu       0                0         (0/0/0)             0


ash# format /dev/did/rdsk/d9s2
selecting /dev/did/rdsk/d9s2

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 128 sec 128>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =  128
nsect       =  128
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264      134.88GB    (17265/0/0) 282869760
  1        usr    wm   17265 - 34529      134.88GB    (17265/0/0) 282869760
  2     backup    wu       0 - 34529      269.77GB    (34530/0/0) 565739520
  3 unassigned    wu       0                0         (0/0/0)             0
```

```
4 unassigned    wu       0                0         (0/0/0)            0
5 unassigned    wu       0                0         (0/0/0)            0
```

```
 6 unassigned    wu       0                0         (0/0/0)            0
 7 unassigned    wu       0                0         (0/0/0)            0
```

The format(1M) command reveals the space available on a device, but it does not reveal whether a disk is mirrored or striped. The above code example's format(1M) output reveals the following information that is used during creation of the mcf file shown in the next code example:

- Output for devices d4s0 and d6s0 shows 16.86 Gbytes each. These devices are assigned equipment numbers 501 and Equipment Number 502, respectively, in the mcf file. They are the appropriate size to use for metadata slices.
- Output for devices d8s0 and d9s0 shows 134.88 Gbytes each. These devices are assigned equipment number 503 and Equipment Number 504, respectively, in the mcf file. They are the are the appropriate size to be used for storing data.

3. Make an ma entry for the file system.
   In this line entry, include the shared keyword in the Additional Parameters field.
4. Make an mm entry listing the partitions that constitute the metadata for the qfs1 file system.
   Put the file system's mm devices on mirrored (RAID-1) disks. The mm devices should constitute about 10% of the space allocated for the entire file system.
5. Make a series of mr entries listing the partitions that constitute the file data for the qfs1 file system.
   The following code example shows the resulting mcf file.

```
#Equipment            Eq   Eq     Family   Additional
#Identifier           Ord  Type   Set      Parameters
#-------------------- ---  ----   ------   ----------
#
# Family Set sqfs1 (shared FS for SunCluster)
#
sqfs1                500   ma    sqfs1    shared
sqfs1                500   ma    sqfs1    shared
/dev/did/dsk/d4s0    501   mm    sqfs1    -
/dev/did/dsk/d6s0    502   mm    sqfs1    -
/dev/did/dsk/d8s0    503   mr    sqfs1    -
/dev/did/dsk/d9s0    504   mr    sqfs1    -
```

Administering File System Quotas

Contents

Administering File System Quotas

About File System Quotas

File system quotas control the amounts of online and total disk space that can be consumed by a specific user, by a group of users, or by a site-determined group of users called an admin set.

Quotas help control the size of a file system by limiting the amount of space and the number of inodes that each user can consume. Quotas can be especially useful on file systems that contain user home directories. After quotas are enabled, you can monitor usage and adjust the quotas as needs change.

A file system provides a user with blocks for data and inodes for files. Each file uses one inode, and file data is stored in a disk allocation unit (DAU). DAU sizes are determined at the time the file system is created. Quotas account for disk usage in multiples of 512 bytes.

The following subsections provide background information about using quotas.

Types of Quotas, Quota Files, and Quota Records

You can set quotas according to user ID, group ID, or an administrator's site-specific grouping. This site-specific grouping is called an admin set ID. You can use an admin set ID, for example, to identify a collection of users working on a project for which file system quotas are imposed.

Quotas are enabled when the `quota` mount option is in effect and the system detects the presence of one or more quota files in the file system's root directory. The `quota` mount option is enabled by default. If you mount the file system with `noquota` in effect, quotas are disabled. For more information about mount options, see the `mount_samfs`(1M) man page.

Each quota file contains a sequence of records. Record zero is the record for the system administrator's quotas and resource usage. System administrator quotas are never enforced, but you can use any record, including the system administrator's record, as a template for subsequent records in the quota file. For more information about this practice, see How to Enable or Change Limits for Users, Groups, or Admin Sets Using an Existing Quota File.

Record one is the record in the quota file for user one, group one, or admin set ID one, depending on the type of quota file. You can edit record one and all subsequent records in order to set different quotas for different users. The following table shows the quota file names and the quotas they enable in `/root`.

Table – Quota File Names

Quota File Name	Quota Type
`.quota_u`	UID (system user ID)
`.quota_g`	GID (system group ID)
`.quota_a`	AID (system admin set ID)

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

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

(highest-ID + 1) x 128 = xx / 4096 = zero quota file size

Soft Limits and Hard Limits

You can set both soft and hard limits. A hard limit specifies a fixed amount of system resources available for use, which the system never allows a user to exceed. A soft limit specifies a level of system resource use that can be exceeded temporarily, up to the hard limit. The soft limit is never larger than the hard limit.

If a user attempts to allocate resources beyond the hard limit, the operation is aborted. In this case, the operation fails and generates an EDQUOT error.

After a user exceeds a soft limit, a timer starts, and the user enters a grace period. While the timer is ticking, the user is allowed to operate above the soft limit. After the user goes below the soft limit, the timer is reset. If the grace period ends and the timer stops without the user's having gone below the soft limit, the soft limit is then enforced as a hard limit.

For example, assume that a user has a soft limit of 10,000 blocks and a hard limit of 12,000 blocks. If the user's block usage exceeds 10,000 blocks and the timer exceeds the grace period, this user is no longer able to allocate more disk blocks on that file system until usage drops below the 10,000-block soft limit.

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

Disk Blocks and File Limits

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

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

Enabling Quotas

You can enable quotas through a process that includes editing system files, creating quota files, and entering various quota commands.

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

Guidelines for Setting Up Quotas

Before you enable quotas, you should determine how much disk space and how many inodes to allocate to each user. If you want to be sure that the total file system space is never exceeded, you can divide the total size of the file system by the number of users. For example, if three users share a 100-megabyte slice and have equal disk space needs, you could allocate 33 megabytes to each. In environments in which not all users are likely to reach their limits, you might want to set individual quotas so that they add up to more than the total size of the file system. For example, if three users shared a 100-megabyte slice, you could allocate 40 megabytes to each.

You can use the following quota commands, in the formats shown, for displaying quota information:

- The `squota`(1) command is for end users. It enables them to retrieve quota information for themselves on a user, group, or admin set basis.
- The `samquota`(1M) command is for system administrators. It enables you to retrieve quota information or to set quotas. Use the `-U`, `-G`, and `-A` options for a user, a group, or an admin set, respectively. The following example shows this.

Example – Using **`samquota`**(1M) to Retrieve Information

The first command displays a user quota. The second command shows a group quota. The last command shows the quota for an admin set.

```
# samquota -U janet </mount-point>
# samquota -G pubs </mount-point>
# samquota -A 99 </mount-point>
```

How to Configure a New File System to Use Quotas

Use this procedure if you are creating a new file system and no files currently reside in the file system. To configure an existing file system to use quotas, see To Configure an Existing File System to Use Quotas.

Before you start this procedure, make sure that you do not have the `noquota` mount option specified in your `samfs.cmd` or `/etc/vfstab` files.

1. Become superuser.
2. Create the file system.
   Either follow the steps outlined in the Sun SAM-QFS Installation and Upgrade Guide, or use the examples in Configuration Examples to create the `mcf` file, create the mount point, initialize the file system, and so on .
3. Use the `mount`(1M) command to mount the file system.
   For example:

```
# mount /qfs1
```

4. Use the dd(1M) command to create the quota files.
   The arguments to this command depend on the type of quota you are creating, as follows:
   - To create admin set quotas, use the following command:

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

   - To create group quotas, use the following command:

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

   - To create user quotas, use the following command:

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

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

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

   ```
   # umount /qfs1
   ```

   The file system must be unmounted so it can be remounted and have its quota files read at mount time. For more information, see the `umount`(1M) man page.

6. Use the `samfsck`(1M) command to perform a file system check.
   In the following example, the -F option resets the in-use values in the quota files:

   ```
   # samfsck -F qfs1
   ```

7. Use the `mount`(1M) command to remount the file system.
   The system enables quotas when it detects the presence of one or more quota files in the `root` directory.

   > Note -
   > You do not need to include the `quota` mount option in the `/etc/vfstab` or `samfs.cmd` file. The `quota` mount option is enabled by default with the `mount`(1M) command, and quotas are enabled automatically when the system detects the presence of quota files.

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

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

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

How to Configure an Existing File System to Use Quotas

Use this procedure if you are creating quotas for a file system that is already populated with files. If you are configuring a new file system to use quotas, see To Configure a New File System to Use Quotas.

Before you start this procedure, make sure that you do not have the `noquota` mount option specified in your `samfs.cmd` or `/etc/vfstab` files.

1. Use the su(1) command to become superuser.
2. Use the mount(1M) command to examine the /etc/mnttab file and ensure that the file system is mounted:

```
# mount
```

   Make sure that the file system is listed in the mount list that is displayed.

3. Use the cd(1) command to change to the root directory of the file system for which quotas are to be enabled. For example:

```
# cd /oldfs1
```

4. Use the ls(1) -a command to retrieve the list of files in this directory and verify that quotas do not already exist on the file system.
   If any of the following files are present, quotas have been enabled for this file system: .quota_u, .quota_g, .quota_a.
   If any quota type is established for a file system, you can establish any other quota type later. Be careful not to modify existing quota files when adding new ones.
5. If the quota files do not exist for the types of quotas you wish to enforce, use the dd(1M) command to create the quota files.
   Determine the highest existing ID numbers of the types of quotas you wish to enforce. Make the initial, zero, quota files large enough to hold the records for those IDs; each quota file record requires 128 bytes.
   For example, if you want to enable admin set quotas, and the highest admin set ID in use on the file system is 1024, the calculation is as follows:
   (1024 + 1) x 128 = 131200
   131200/4096 = 32.031...
   Use the following command:

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

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

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

```
# umount /oldfs1
```

   The file system must be unmounted so it can be remounted and have its quota files read at mount time. For more information about unmounting a file system, see Unmounting a File System.

7. Use the samfsck(1M) -F command to perform a file system check. This command updates records allocated in the quota files with current usage information. For example:

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

8. Use the mount(1M) command to remount the file system in which the quota files have been created.
   The system enables quotas when it detects the presence of one or more quota files in the /root directory.
   You do not need to include the quota mount option in the /etc/vfstab or samfs.cmd file. The quota mount option is enabled by default with the mount(1M) command, and quotas are enabled automatically when the system detects the presence of quota files.

   > Note -
   > If quota files are present and if the file system is mounted with quotas disabled, the quota records become inconsistent with actual usages when blocks or files are allocated or freed. If a file system with quotas is mounted and run with quotas disabled, run the samfsck(1M) -F command to update the quota file usage counts before again remounting the file system with quotas enabled.

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

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

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

How to Assign Admin Set IDs to Directories and Files

1. Use the su(1) command to become superuser.
2. Set the admin IDs. Use the samchaid(1M) command to change the admin set IDs for the directory or file, as follows:
   - To set IDs for a file or directory, specify the directory name or path. For example:

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

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

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

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

Setting Infinite Quotas

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

How to Set an Infinite Quota

- Use the samquota(1M) command to set the quota limit to zero. For example:

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

  You can set infinite quotas for particular users, groups, or admin set IDs by setting zero values for all hard and soft limits. The following example shows how to set infinite quotas.

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

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

Enabling Default Quota Values

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

How to Enable Default Quota Values for Users, Groups, or Admin Sets

- Use the samquota(1M) command. The following command sets default quotas for all admin set IDs:

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

The preceding command sets any user's uninitialized admin set quota limits as follows:

- The soft online block limit (-b limit:s) is set to 12,000 blocks.
- The hard online block limit (-b limit:h) is set to 15,000 blocks.
- The total soft block limit (-b limit:s:t) is set to 12 gigablocks.
- The total hard block limit (-b limit:h:t) is set to 15 gigablocks.
- The soft file limit (-f limit:s) is set to 1000 files.
- The hard file limit (-f limit:h) is set to 1200 files.
- The grace period (-t limit) is set to one week.

> Note -
> If a quota record already exists, the existing values remain in effect. This occurs, for example, if the admin group already has blocks assigned to it.

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

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

Enabling Limits

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

How to Enable Limits for Users, Groups, or Admin Sets

- Use the `samquota`(1M) command. For example, the following commands enable limits for users, groups, and admin sets, respectively.

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

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

How to Enable or Change Limits for Users, Groups, or Admin Sets Using an Existing Quota File

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

1. Retrieve a quota file and direct the output to a temporary file.
Use the `samquota`(1M) command with the `-e` option and with one or more of the following options: `-U` userID, `-G` groupID, or `-A` adminsetID.
The following example shows how to retrieve the `quota.group` file to use as a template.

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

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

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

2. Save the file and exit the editor.
3. To apply the changes made in the editor, execute the file using the shell. For example:

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

The -x option directs the shell to echo the commands it executes. You can omit the -x option if desired.

Checking Quotas

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

How to Check for Exceeded Quotas

1. Become superuser.
2. Display the quotas in effect for mounted file systems.
   - Display user quotas, using the following command:

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

   For userID, specify the numeric user ID or user name of the account whose quotas are being examined.
   For file, specify a file system for the specified user, group, or admin set. The file argument can also be the name of any file in the file system. Typically, file is the name of the root directory of the file system.
   Example 1. The following example retrieves user hm1259 quota statistics in the sam6 file system on the server and displays output indicating that this user is not exceeding the quota.

   ```
   # samquota -U hm1259 /sam6

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

   Example 2. The following example retrieves user memil quota statistics in all mounted Sun QFS file systems and displays output indicating that this user is exceeding the quota. Note the plus sign (+) in the Blocks row of the output. In the case of the soft quota limit also being exceeded, a plus sign is also displayed in the Files row.

```
# samquota -U memil

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

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

- Display group quotas, using the following command:

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

For groupID, specify the numeric group ID or the group name for the group of users whose quotas are being examined. For file, specify a file system for the specified group. The file argument can also be the name of any file in the file system. Typically, file is the name of the root directory of the file system.
For example, the following command retrieves user quota statistics for the group `turtles` in the `qfs3` file system:

```
# samquota -G turtles /qfs3
```

- Display admin set quotas, using the following command:

```
# samquota -A <adminsetID> [ <file> ]
```

For adminsetID, specify the numeric admin set ID of the administrator set whose quotas are being examined. For file, specify a file system for the specified admin set. The file argument can also be the name of any file in the file system. Typically, file is the name of the root directory of the file system.
For example, the following command retrieves user quota statistics for the admin set 457 in all mounted file systems:

```
# samquota -A 457
```

Changing and Removing Quotas

You can change quotas to adjust the amount of disk space or number of inodes allocated to users. You can also remove quotas from users or from an entire file system. The following subsections describe how to change and remove quotas.

How to Change the Grace Period

You can use the `samquota`(1M) command to change the soft time limit grace period.

1. Retrieve quota statistics for a user, group, or admin set.
   See How to Check for Exceeded Quotas for instructions. The following example retrieves information about group `sam` and shows that

this group is over its soft limit.

```
# samquota -G sam /sam6

                         Online Limits                Total Limits
Type    ID    In Use     Soft     Hard    In Use     Soft     Hard
/sam6
Files  group   101        32     2000     2000        32     2000     2000
Blocks group   101     41888*   40000 60000000     43208 60000000 60000000
Grace period                    1w                            1w
---> Online soft limits under enforcement (since 30s ago)
```

2. Examine the output and determine the new limits.
3. Change the soft time limit grace period. The following example shows the `samquota`(1M) command options to use.

```
# samquota -U <userID> -t <interval> <file>
# samquota -G <groupID> -t <interval> <file>
# samquota -A <adminID> -t <interval> <file>
```

The arguments for these commands are as follows:

- userID is the numeric user ID or user name of the user whose quotas are being changed.
- groupID is the numeric group ID or the group name for the group of users whose quotas are being changed.
- adminID is the numeric admin set ID of the administrator set whose quotas are being changed.
- interval is the duration of the grace period. Specify an integer to indicate the quantity and specify a unit of time, if desired. The default unit is `s`, which indicates seconds. You can specify `w` for weeks, `d` for days, `h` for hours, or `m` for minutes.
- file is the file system for the specified user, group, or admin set. The file argument can also be the name of any file in the file system. Typically, file is the name of the root directory of the file system

Example – Changing the Grace Period

1. To change the grace period for user `memil`, first verify the quotas:

```
# samquota -U memil /sam6

                         Online Limits                Total Limits
Type    ID    In Use     Soft     Hard    In Use     Soft     Hard
/sam6
Files   user 130967         4      500      750         4      500      750
Blocks  user 130967     41016+   40000    50000     41016    50000    50000
Grace period                    3d                            0s
---> Warning:  online soft limits to be enforced in 2d23h59m7s
```

2. Shorten the grace period:

```
# samquota -U memil -t 1d /sam6
```

3. Verify the new quotas.

```
# samquota -U memil /sam6
                         Online Limits                Total Limits
Type    ID    In Use     Soft     Hard    In Use     Soft     Hard
/sam6
Files   user 130967         4      500      750         4      500      750
Blocks  user 130967     41016+   40000    50000     41016    50000    50000
Grace period                    1d                            0s
---> Warning:  online soft limits to be enforced in 23h58m31s
```

Changing the Grace Period Expiration

If a user has exceeded the soft quota limit, changing the grace period itself does not modify the expiration timer of any grace periods that have already started. If the grace period is already in effect, you can use the `samquota`(1M) command to modify the grace period in one of the following ways:

- Clear the grace period timer - The next time the user allocates a file or block while still over a soft limit, the grace period timer is reset and the grace period restarts.
  The following example shows the command used to clear the timer so it starts counting the next time a user in group `sam` attempts to allocate a block or file in `/sam6`.

Example – Clearing the Timer

```
# samquota -G sam -x clear /sam6

Setting Grace Timer:  continue? y

# samquota -G sam  /sam6

                                 Online Limits                Total Limits
        Type    ID    In Use      Soft     Hard    In Use     Soft     Hard
/sam6
Files  group   101        32      2000     2000        32     2000     2000
Blocks group   101     41888+    40000 60000000     43208 60000000 60000000
Grace period                     1w                          1w
---> Warning:  online soft limits to be enforced in 6d23h59m56s
```

- Reset the grace period timer - When an expiration period is reset, the timer is reset and the grace period restarts.
  The following example resets the grace period.

Example – Resetting the Grace Period Timer

```
# samquota -G sam -x reset /sam6

Setting Grace Timer:  continue? y

# samquota -G sam  /sam6

                                 Online Limits                Total Limits
        Type    ID    In Use      Soft     Hard    In Use     Soft     Hard
/sam6
Files  group   101        32      2000     2000        32     2000     2000
Blocks group   101     41888     40000 60000000     43208 60000000 60000000
Grace period                     1w                          1w
---> Warning:  online soft limits to be enforced in 6d23h59m52s
```

- Set the grace period to a value - The timer is set to a value, and it starts counting down immediately from that value. There are no restrictions on this value. The value can be larger than the grace period. The following example sets a very long expiration period.

Example – Setting a Very Long Grace Period

```
# samquota -G sam -x 52w /sam6

Setting Grace Timer:  continue? y

# samquota -G sam  /sam6

                                 Online Limits                Total Limits
        Type    ID    In Use      Soft      Hard    In Use     Soft      Hard
/sam6
Files  group   101        32      2000      2000        32     2000      2000
Blocks group   101     41888+    40000 60000000     43208 60000000 60000000
Grace period                    1w                           1w
---> Warning:  online soft limits to be enforced in 51w6d23h59m54s
```

- Expire the grace period timer - The timer is set to expire immediately. The following example expires the grace period.

Example – Expiring the Grace Period Timer

```
# samquota -G sam -x expire /sam6

Setting Grace Timer:  continue? y

# samquota -G sam  /sam6

                                 Online Limits                Total Limits
        Type    ID    In Use      Soft      Hard    In Use     Soft      Hard
/sam6
Files  group   101        32      2000      2000        32     2000      2000
Blocks group   101     41888     40000 60000000     43208 60000000 60000000
Grace period                    1w                           1w
---> Online soft limits under enforcement (since 6s ago)
```

How to Inhibit Additional File System Resource Allocations

When the file system detects that quota values are not consistent for a user, group, or admin set, it prevents that user, group, or admin set from using any more system resources. You can inhibit file system resource allocations by creating inconsistent quota values. For example, you can inhibit further allocation if the hard block or file limits are lower than the soft block or file limits, or if a user's soft limit is larger than the user's hard limit.

The file system treats an inconsistent quota setting as a special quota. You can set inconsistent quota values into record zero of the user, group, or admin set ID quota files, and from there they can become the default values for new users, groups, or admin set IDs.

The following procedure shows how to inhibit further system resource allocations for a user, group, or admin set.

1. Become superuser.
2. Obtain, save, and examine current quota information.
   The following example shows how to retrieve current group quota information for group sam and write it to a backup file.

```
# samquota -G sam -e /sam6 | & tee restore.quota.sam

# Type  ID
#                Online Limits                          Total   Limits
#            soft            hard                    soft            hard
# Files
# Blocks
# Grace Periods
#
samquota -G 101 \
      -f     2000:s:o -f     2000:h:o         -f     2000:s:t -f     2000:h:t \
      -b    40000:s:o -b 60000000:h:o         -b 60000000:s:t -b 60000000:h:t \
                 -t  1w:o                                 -t 1w:t \
      -x 51w6d23h59m:o                       -x clear     /sam6
```

To obtain quota information about a user quota, specify the -U userID option in place of the -G option. To obtain quota information about an admin set quota, specify the -A adminID option in place of the -G option.

3. Use the samquota(1M) command to set soft quotas to nonzero quotas and hard quotas to zero quotas.
   The following command sets the quotas for group sam to be inconsistent:

```
# samquota -G sam -f 1:s -f 0:h -b 1:s -b 0:h /sam6
```

   To make the quotas for users or admin sets inconsistent, specify the -U userID or{{-A}} adminID option in place of the -G option.

4. Use the samquota(1M) command to verify your changes, as in the following example:

```
# samquota -G sam /sam6

Online Limits                Total Limits
Type     ID    In Use     Soft     Hard    In Use     Soft     Hard
/sam6
Files  group   101        32!        1         0        32!        1        0
Blocks group   101     41888!        1         0     43208!        1        0
Grace period                  1w                          1w
---> Quota values inconsistent; zero quotas in effect.
```

   In the preceding output, a zero quota is in effect. The exclamation point characters (!) indicate the over-quota condition in the output.

5. Use the sh(1) and samquota(1M) commands to restore the group's quota to what it was before the file/block allocation was inhibited and then to verify the changed quotas. The following example shows these commands.

```
# sh restore.quota.sam
Setting Grace Timer:  continue? y
Setting Grace Timer:  continue? y
# samquota -G sam /sam6
Online Limits                Total Limits
Type     ID    In Use     Soft     Hard    In Use     Soft     Hard
/sam6
Files  group   101        32     2000      2000        32     2000     2000
Blocks group   101     41888+    40000 60000000     43208 60000000 60000000
Grace period                  1w                          1w
---> Warning:  online soft limits to be enforced in 6d23h59m54s
```

   To perform this operation on a user quota, specify the -U userID option in place of the -G option. To perform this operation on an admin set quota, specify the -A adminID option in place of the -G option.

How to Remove the Quotas for a File System

To remove or disable quotas for a file system, disable quotas in the mount process.

1. Become superuser.
2. (Optional) Use a text editor to add the noquota mount option to the /etc/vfstab or samfs.cmd file.
   As an alternative, you can specify noquota as an option when you issue the mount command. See Step 4.
3. If the file system is mounted, use the umount(1M) command to unmount the file system. For example:

```
# umount /myfs
```

   If you have difficulty unmounting the file system, see Unmounting a File System.

4. Remount the file system using the mount(1M) command.

If you did not perform Step 2, include the noquota option with the mount(1M) command. For example:

```
# mount -o noquota /myfs
```

5. Manage the quota files by doing one of the following:
   - If you expect to reinstate the quota feature later and therefore do not want to destroy the quota files, unmount the file system, run the samfsck(1M) command with its -F option on the file system, and remount the file system with the noquota option removed.
   - If you do not expect to reinstate the quota feature or if you want to reclaim the space consumed by the quota files, use the rm(1) command to remove the .quota_u, .quota_g, and .quota_a files. For example:

     ```
     # rm /myfs/.quota_[agu]
     ```

How to Correct Quotas

1. Become superuser.
2. If the file system is mounted, use the umount(1M) command to unmount the file system. For example:

   ```
   # umount /myfs
   ```

   If you have difficulty unmounting the file system, see Unmounting a File System.

3. Perform a file system check. The samfsck(1M) command updates records allocated in the quota files with correct, current usage information. For example:

   ```
   # samfsck -F myfs
   ```

4. Use the mount(1M) command to remount the file system. For example:

   ```
   # mount /myfs
   ```

Advanced File System Topics

Contents

Advanced File System Topics

This section discusses advanced topics that are beyond the scope of basic system administration and usage.

Using Daemons, Processes, and Tracing

It is useful to have an understanding of system daemons and processes when you are debugging. This section describes the Sun SAM and Sun QFS daemons and processes. It also provides information about daemon tracing.

Daemons and Processes

All daemons are named in the form sam-daemon_named. Processes are named in a similar manner, except that processes do not end in the lowercase letter d.

The following table shows some of the daemons and processes that can run on your system. Others, such as `sam-genericd` and `sam-catserverd`, might also be running, depending on system activities.

Table – Daemons and Processes

Process	Description
sam-amld	Initializes the Sun SAM automated library daemons: `sam-catserverd`, `sam-scannerd`, and `sam-robotsd`.
`sam-archiverd`	Automatically archives Sun SAM files. This process runs as long as the Sun SAM file system is mounted.
`sam-catserverd`	Keeps track of media in Sun SAM and SAM-QFS library catalogs.
`sam-fsd`	Master daemon.
`sam-rftd`	Transfers data between multiple Sun SAM host systems.
`sam-robotsd`	Starts and monitors automated library media changer control daemons.
`sam-scannerd`	Monitors all manually mounted removable media devices. The scanner periodically checks each device for inserted archive media cartridges.
`sam-sharefsd`	Invokes the Sun QFS shared file system daemon.
`sam-releaser`	Attempts to release disk space occupied by previously archived files on Sun SAM file systems until a low-water mark is reached. The releaser is started automatically when a high-water mark is reached on disk cache and stops when it has finished releasing files. This is a process, not a daemon.
`sam-stagealld`	Controls the associative staging of Sun SAM files.
`sam-stagerd`	Controls the staging of Sun SAM files.
`sam-rpcd`	Controls the remote procedure call (RPC) application programming interface (API) server process.
sam-robotsd	Starts and monitors the execution of the media changer library control daemons for Sun SAM.

When you run the software, SMF starts the `sam-fsd` daemon. It should restart automatically in case of failure.

In a shared file system, a `sam-fsd` daemon is always active. In addition, one `sam-sharefsd` daemon is active for each mounted shared file system.

When a shared file system is mounted, the software starts a shared file system daemon (`sam-sharefsd`). TCP sockets are used to communicate between the server and client hosts. All clients that connect to the metadata server are validated against the hosts file.

> Note -
> See the `hosts.fs`(4) man page for more information about the hosts file.

The `sam-sharedfsd` daemon on the metadata server opens a listener socket on the port named `sam-qfs`. During the Sun QFS installation process, the `sam-qfs` entry is automatically added to `/etc/services` file. Do not remove this entry. In addition, the shared file system port is defined in the `/etc/inet/services` file as port number 7105. Verify that this port does not conflict with another service.

> **Note -**
> Before the Sun QFS 4U2 release, one port per file system was required. You can remove these entries from your file.

All metadata operations, block allocation and deallocation, and record locking are performed on the metadata server. The `sam-sharefsd` daemon does not keep any information. Hence, it can be stopped and restarted without causing any consistency problems for the file system.

Trace Files

Several processes can write messages to trace files. These messages contain information about the state and progress of the work performed by the daemons. The messages are primarily used by Sun Microsystems staff to improve performance and diagnose problems. The message content and format are subject to change from release to release.

Trace files can be used in debugging. By default, trace files are not enabled. You can enable trace files by editing the `defaults.conf` file. You can enable tracing for all processes, or you can enable tracing for individual processes. For information about the processes that you can trace, see the `defaults.conf`(4) man page.

By default, trace files are written to the `/var/opt/SUNWsamfs/trace` directory. In that directory, the trace files are named for the processes (`archiver`, `catserver`, `fsd`, `ftpd`, `recycler`, `sharefsd`, and `stager`). You can change the names of the trace files by specifying directives in the `defaults.conf` configuration file. You can also set a limit on the size of a trace file and rotate your tracing logs. For information about controlling tracing, see the `defaults.conf`(4) man page.

Trace File Content

Trace file messages contain the time and source of the message. The messages are produced by events in the processes. You can select the events by using directives in the `defaults.conf` file.

The default events are as follows:

- Customer notification `syslog` or `notify` file messages
- Nonfatal program errors
- Fatal `syslog` messages
- Process initiation and completion
- Other miscellaneous events

You can also trace the following events:

- Memory allocations
- Interprocess communication
- File actions
- Operator messages
- Queue contents when changed
- Other miscellaneous events

The default message elements (program name, process ID (PID), and time) are always included and cannot be excluded. Optionally, the messages can also contain the following elements:

- The date (The time is always included.)
- The source file name and line number
- The event type

Trace File Rotation

To prevent trace files from growing indefinitely, the `sam-fsd` daemon monitors the size of the trace files and periodically executes the following command:

```
/opt/SUNWsamfs/sbin/trace_rotate
```

This script moves the trace files to sequentially numbered copies. You can modify this script to suit your operation. Alternatively, you can provide this function using `cron`(1) or some other facility.

Determining Which Processes Are Being Traced

To determine which processes are being traced currently, enter the sam-fsd(1M) command at the command line. The following example shows the output from this command.

Example – **sam-fsd**(1M) Command Output

```
# sam-fsd
Trace file controls:
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
             cust err fatal misc proc date
             size    10M  age 0
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
             cust err fatal ipc misc proc queue date module
             size    10M  age 0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
             cust err fatal misc proc date
             size    10M  age 0
sam-dbupd     /var/opt/SUNWsamfs/trace/sam-dbupd
             cust err fatal misc proc date
             size    10M  age 0
sam-fsalogd   /var/opt/SUNWsamfs/trace/sam-fsalogd
             cust err fatal misc proc date
             size    10M  age 0
sam-fsd       /var/opt/SUNWsamfs/trace/sam-fsd
             cust err fatal misc proc date
             size    10M  age 0
sam-rftd      /var/opt/SUNWsamfs/trace/sam-rftd
             cust err fatal misc proc date
             size    10M  age 0
sam-recycler  /var/opt/SUNWsamfs/trace/sam-recycler
             cust err fatal ipc misc proc date module type
             size    10M  age 0
sam-nrecycler /var/opt/SUNWsamfs/trace/sam-nrecycler
             cust err fatal misc proc date
             size    10M  age 0
sam-sharefsd  /var/opt/SUNWsamfs/trace/sam-sharefsd
             cust err fatal misc proc date
             size    10M  age 0
sam-stagerd   /var/opt/SUNWsamfs/trace/sam-stagerd
             cust err fatal ipc misc proc date module
             size    10M  age 0
sam-serverd   /var/opt/SUNWsamfs/trace/sam-serverd
             cust err fatal misc proc date
             size    10M  age 0
sam-clientd   /var/opt/SUNWsamfs/trace/sam-clientd
             cust err fatal misc proc date
             size    10M  age 0
fsmgmt        /var/opt/SUNWsamfs/trace/fsmgmt
             cust err fatal misc proc date
             size    10M  age 0
sam-shrink    /var/opt/SUNWsamfs/trace/sam-shrink
             cust err fatal misc proc date
             size    10M  age 0
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
#
```

For more information about enabling trace files, see the defaults.conf(4) man page and the sam-fsd(1M) man page.

Using the **setfa**(1) Command to Set File Attributes

Sun QFS file systems enable end users to set performance attributes for files and directories. Applications can enable these performance features on a per-file or per-directory basis. The following sections describe how the application programmer can use these features to select file attributes for files and directories, to preallocate file space, to specify the allocation method for the file, and to specify the disk stripe width.

For more information about implementing the features described in the following subsections, see the setfa(1) man page.

Selecting File Attributes for Files and Directories

The setfa(1) command sets attributes on a new or existing file. The file is created if it does not already exist.

You can set attributes on a directory as well as a file. When using setfa(1) with a directory, files and directories created within that directory inherit the attributes set in the original directory. To reset attributes on a file or directory to the default, use the{{– d}} (default) option. When the{{– d}} option is used, attributes are first reset to the default and then other attributes are processed.

Preallocating File Space

An end user can preallocate space for a file. This space is associated with a file so that no other files in the file system can use the disk addresses allocated to this file. Preallocation ensures that space is available for a given file, which avoids a file-system-full condition. Preallocation is assigned at the time of the request rather than when the data is actually written to disk.

Note that space can be wasted by preallocation of files. If the file size is less than the allocation amount, the kernel allocates space to the file from the current file size up to the allocation amount. When the file is closed, space below the allocation amount is not freed.

You can preallocate space for a file by using the setfa(1) command with either the -L or the{{– l}} (lowercase letter L) option. Both options accept a file length as their argument. Use the -L option for an existing file, which can be empty or contain data. Use the -l option for a file that has no data yet. If you use the -l option, the file cannot grow beyond its preallocated limit.

For example, to preallocate a 1-gigabyte file named /qfs/file_alloc, type the following:

```
# setfa -l 1g /qfs/file_alloc
```

After space for a file has been preallocated, truncating a file to 0 length or removing the file returns all space allocated for a file. There is no way to return only part of a file's preallocated space to the file system. In addition, if a file is preallocated with the -l option, there is no way to extend the file beyond its preallocated size in future operations.

Selecting a File Allocation Method and Stripe Width

By default, a file uses the allocation method and stripe width specified at mount time (see the mount_samfs(1M) man page). However, an end user might want to use a different allocation scheme for a file or directory. The user could do this by using the setfa(1) command with the -s (stripe) option.

The allocation method can be either round-robin or striped. The -s option specifies the allocation method and the stripe width, as shown in the following table.

Table – File Allocations and Stripe Widths

-s Option	Allocation Method	Stripe Width	Explanation
0	Round-robin	Not applicable	The file is allocated on one device until that device has no space.
1–255	Striped	1–255 DAUs	The file is striped across all disk devices with this number of DAUs per disk.

The following example shows how to create a file explicitly by specifying a round-robin allocation method:

```
# setfa -s 0 /qfs/100MB.rrobin
```

The following example shows how to create a file explicitly by specifying a striped allocation method with a stripe width of 64 DAUs (preallocation is not used):

```
# setfa -s 64 /qfs/file.stripe
```

Selecting a Striped Group Device

Striped group devices are supported for Sun QFS file systems only.

A user can specify that a file begin allocation on a particular striped group. If the file allocation method is round-robin, the file is allocated on the designated stripe group.

The following example shows `setfa`(1) commands specifying that `file1` and `file2` be independently spread across two different striped groups.

Example – **`setfa`**(1) Commands to Spread Files Across Striped Groups

```
# setfa -g0 -s0 file1
# setfa -g1 -s0 file2
```

This capability is particularly important for applications that must achieve levels of performance that approach raw device speeds. For more information, see the `setfa`(1) man page.

Accommodating Large Files

When manipulating very large files, pay careful attention to the size of disk cache that is available on the system. If you try to write a file that is larger than your disk cache, behavior differs depending on the type of file system that you are using:

- If you are using a non-archiving file system, the system returns an ENOSPC error.
- If you are using an archiving file system, the program blocks, waiting for space that might never exist, because the available disk space is insufficient to handle the request.

If you are operating within an archiving environment and your application must write a file that is larger than the disk cache, you can segment the file with the `segment`(1) command. For more information about the `segment`(1) command, see the `segment`(1) man page or see Using Segmented Files.

Configuring a Multireader File System

The multireader file system consists of a single writer host and multiple reader hosts. The `writer` and `reader` mount options that enable the multireader file system are compatible with Sun QFS file systems only. The mount options are described in this section and on the `mount_samfs`(1M) man page.

You can mount the multireader file system on the single writer host by specifying the{{– o writer}} option with the `mount`(1M) command. The host system with the `writer` mount option is the only host system that is allowed to write to the file system. The `writer` host system updates the file system. You must ensure that only one host in a multireader file system has the file system mounted with the `writer` mount option enabled. If{{– o writer}} is specified, directories are written through to disk at each change and files are written through to disk at close.

> Caution -
> The multireader file system can become corrupted if more than one writer host has the file system mounted at one time. It is the site administrator's responsibility to ensure that this situation does not occur.

You can mount a multireader file system on one or more reader hosts by specifying the{{– o reader}} option with the `mount`(1M) command. There is no limit to the number of host systems that can have the multireader file system mounted as a reader.

A major difference between the multireader file system and a Sun QFS shared file system is that the multireader host reads metadata from the disk, and the client hosts of a Sun QFS shared file system read metadata over the network. The Sun QFS shared file system supports multireader hosts. In this configuration, multiple shared hosts can be adding content while multiple reader hosts are distributing content.

> Note -
> You cannot specify the `writer` option on any host if you are mounting the file system as a Sun QFS shared file system. You can, however, specify the `reader` option. If you want a Sun QFS shared file system client host to be a read-only host, mount the Sun QFS shared file system on that host with the `reader` mount option. In addition, set the `sync_meta` mount option to `1` if you use the `reader` option in a Sun QFS shared file system. For more information about the Sun QFS shared file system, see Configuring a Shared File System . For more information about mount options, see the `mount_samfs`(1M) man page.

You must ensure that all readers in a multireader file system have access to the device definitions that describe the `ma` device. Copy the lines from the mcf file that resides on the primary metadata server host to the mcf files on the alternate metadata servers. After copying the lines, you might need to update the information about the disk controllers because, depending on your configuration, disk partitions might not show up the same way across all hosts.

In a multireader file system environment, the Sun QFS software ensures that all servers accessing the same file system can always access the current environment. When the writer closes a file, the Sun QFS file system immediately writes all information for that file to disk. A reader host can access a file after the file is closed by the writer. You can specify the refresh_at_eof mount option to help ensure that no host system in a multireader file system gets out of sync with the file system.

By default, the metadata information for a file on a reader host is invalidated and refreshed every time a file is accessed. If the data changed, it is invalidated. This includes any type of access, whether through cat(1), ls(1), touch(1), open(2), or other methods. This immediate refresh rate ensures that the data is correct at the time the refresh is done, but it can affect performance. Depending on your site preferences, you can use the mount(1M) command's{{– o invalid=}}n option to specify a refresh rate between 0 seconds and 60 seconds. If the refresh rate is set to a small value, the Sun QFS file system reads the directory and other metadata information n seconds after the last refresh. More frequent refreshes result in more overhead for the system, but stale information can exist if n is nonzero.

> Caution -
> If a file is open for a read on a reader host, there is no protection against that file being removed or truncated by the writer. You must use another mechanism, such as application locking, to protect the reader from inadvertent writer actions.

Understanding I/O Types

The Sun QFS file systems support paged I/O, direct I/O, and switching between the I/O types. The following sections describe these I/O types.

Paged I/O

When paged I/O is used, user data is cached in virtual memory pages, and the kernel writes the data to disk. The standard Solaris OS interfaces manage paged I/O. Paged I/O (also called buffered or cached I/O) is selected by default.

Direct I/O

Direct I/O is a process by which data is transferred directly between the user's buffer and the disk. This means that much less time is spent in the system. For performance purposes, specify direct I/O only for large, block-aligned, sequential I/O.

The setfa(1) command and the sam_setfa(3) library routine both have a{{– D}} option that sets the direct I/O attribute for a file or directory. If applied to a directory, files and directories created in that directory inherit the direct I/O attribute. After the{{– D}} option is set, the file uses direct I/O.

You can also select direct I/O for a file by using the Solaris OS directio(3C) function call. If you use the function call to enable direct I/O, the setting lasts only while the file is active.

To enable direct I/O on a file system basis, do one of the following:

- Specify the{{– o forcedirectio}} option with the mount(1M) command.
- Put the forcedirectio keyword in the mount option column of the /etc/vfstab file, or use it as a directive in the samfs.cmd file.

For more information, see the setfa(1), sam_setfa(3), directio(3C), samfs.cmd(4), and mount_samfs(1M) man pages.

I/O Switching

By default, paged I/O is performed, and I/O switching is disabled. However, the Sun QFS file systems support automatic I/O switching, a process by which a site-defined amount of paged I/O occurs before the system switches automatically to direct I/O.

I/O switching should reduce page cache usage on large I/O operations. To enable I/O switching, use samu(1M), or use the dio_wr_consec and dio_rd_consec parameters as directives in the samfs.cmd file or as options with the mount(1M) command.

For more information about these options, see the mount_samfs(1M) or samfs.cmd(4) man pages.

Related Topics

- Configuring WORM-FS File Systems

Configuring a Shared File System

Contents

Configuring a Shared File System

For information about configuring a shared file system in a Sun Cluster environment, see Configuring Sun QFS File Systems With Solaris Cluster.

Using Shared QFS With NFS

Note -
If you are using NFS v4, you must disable delegations before you can use shared QFS.

Starting with SAM-QFS 5.0 on Solaris 10 OS, the Service Management Facility (SMF) is used to manage the mounting of the file system at boot time. If your file system uses NFS, the exact sequence in which you configure NFS and shared QFS is important. If you do not follow the steps below, either the shared QFS mount or the NFS share will succeed, and the other will fail.

How to Configure Shared QFS With NFS

1. Export the existing NFS configuration to a file.
   The following example exports the configuration into a file `/var/tmp/server.xml`.

   ```
   svccfg export /network/nfs/server > /var/tmp/server.xml
   ```

2. Edit the file that you just exported.
   For example:

   ```
   vi /var/tmp/server.xml
   ```

3. After the local file system dependency listed in the file, add a dependency to mount QFS file systems before you NFS share them.
   For example:

```
<!--
  Must have QFS filesystems mounted before sharing them
-->
   <dependency name='qfs'
   grouping='require_all'
   restart_on='error'
   type='service'>
   <service_fmri value='svc:/network/qfs/shared-mount:default'/>
   </dependency>
```

4. Validate the changes that you made to the file.

```
svccfg validate /var/tmp/server.xml
```

5. Disable NFS.

```
svcadm disable nfs/server
```

6. Delete the existing NFS server configuration.

```
svccfg delete nfs/server
```

7. Import the file that you edited in Step 2 into SMF.

```
svccfg import /var/tmp/server.xml
```

8. Enable NFS.
   NFS uses the updated file and reads the QFS dependency information.

```
svcadm enable nfs/server
```

9. Confirm that the dependency is applied.

```
svcs -d svc:/network/nfs/server:default
```

Mounting and Unmounting Shared File Systems

When you mount or unmount a shared file system, the order in which you mount or unmount the metadata server and the clients is important.

For failover purposes, the mount options should be the same on the metadata server and all potential metadata servers. For example, you can create a `samfs.cmd` file that contains the mount options and copy that file to all of the hosts.

For more information about mounting shared file systems, see Mount Options in a Shared File System and the `mount_samfs`(1M) man page. For more information about mounting and unmounting file systems, see Performing Operations.

How to Mount a Shared File System

1. Become superuser on the metadata server and on all the client hosts.
2. Use the `mount`(1M) command to mount the metadata server.
   Mount the file system on the metadata server before you mount the file system on any client hosts. For example:

```
# mount -F samfs qfs1 /qfs -o shared
```

> Tip -
> If the mount information has been placed in `/etc/vfstab`, you can use the simpler command:
> `# mount qfs1`
> For information about the `/etc/vfstab` file and other mount options, see Setting Up Mount Parameters.

3. Use the `mount`(1M) command to mount the client hosts.
   You can mount the file system on the client hosts in any order. For more information about the `mount`(1M) command, see the `mount` (1M) man page.

How to Unmount a Shared File System

> Note -
> If the file system is shared through NFS or SAMBA, unshare the file system before you unmount it.

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

```
client# umount /samqfs
```

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

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

2. Unmount the file system on the metadata server:

```
metaserver# umount /samqfs
```

   At unmounting time, several conditions can be present in a file system that may require you to issue the `umount`(1M) command a second time.

> Note -
> If the file system still does not unmount, use `unshare`(1M), `fuser`(1M), or another command in conjunction with the `umount`(1M) command.

   For more information on unmounting procedures, see the `umount`(1M) man page and Unmounting a File System.
   You can also use the `-o await_clients #` flag with the `umount` command. This causes the unmount process to wait a specified number of seconds (#) for clients to unmount. At the end of the waiting period, or as soon as all clients have unmounted, the unmount proceeds. If this argument is specified for a non-shared file system, or if the host is not the metadata server for the shared file system, the option will be ignored.
   This flag can also be used in conjunction with the `-f` option. In this case, the software will wait for the specified time period before forcing the unmount.

Adding or Removing a Client Host

The following subsections provide instructions for adding and removing client host systems in a shared file system.

How to Add a Client Host to a Shared File System

You can add a client host to a shared file system after you have configured and mounted the file system on all participants.

1. Become superuser on the metadata server.
2. Use the samsharefs(1M) command to retrieve the current shared file system information and write it to an editable file.
   - If the shared file system is mounted, issue the samsharefs(1M) command on the current metadata server. For example:

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

   - If the shared file system is not mounted, issue the samsharefs(1M) command with its -R option from the metadata server or from any of the potential metadata servers. For example:

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

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

3. Use vi(1) or another editor to open the shared file system information file.
   The following code example shows this step.

   ```
   # vi /etc/opt/SUNWsamfs/hosts.sharefs1
   # File /etc/opt/SUNWsamfs/hosts.sharefs1
   # Host    Host IP    Server    Not    Server
   # Name    Addresses    Priority    Used    Host
   # ----    -------------------------------    --------    ----    -----
   titan    172.16.0.129     1    -    server
   tethys    172.16.0.130     2    -
   mimas    mimas     -    -
   dione    dione     -    -
   ```

4. Use the editor to add a line for the new client host.
   The following code example shows the file after addition of the line for helene as the last line.

   ```
   # File /etc/opt/SUNWsamfs/hosts.sharefs1
   # Host    Host IP           Server     Not     Server
   # Name    Addresses         Priority   Used    Host
   # ----    ---------------   --------   ----    -----
   titan     172.16.0.129        1          -     server
   tethys    172.16.0.130        2          -
   mimas     mimas               -          -
   dione     dione               -          -
   helene    helene              -          -
   ```

5. Use the samsharefs(1M) command to update the current information in the binary file.
   The options to use with this command, and the system from which this command is issued, depend on whether the Sun QFS shared file system is mounted, as follows:
   - If the file system is mounted, issue the samsharefs(1M) -u command from the current metadata server. For example:

     ```
     # samsharefs -u sharefs1
     ```

   - If the file system is not mounted, issue the samsharefs(1M) -R -u command from the active metadata server or from any of the potential metadata servers. For example:

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

     The client host helene is now recognized.

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

7. Use the `format`(1M) command to verify the presence of client host disks.
8. Update the `mcf` file on the client host.
   Before a host system can access or mount a shared file system, it must have that file system defined in its `mcf` file. The `mcf` file must be updated to match all client hosts in the shared file system. The file system and disk declaration information must have the same data for the Family Set Name, Equipment Number, and Equipment Type as the configuration on the metadata server. The `mcf` files on the client hosts must also include the shared keyword. The device names, however, can change, since controller assignments will probably differ from host to host.
   For information on how to edit the `mcf` file, see Updating the mcf file in a Sun QFS Shared Environment.
9. Issue the `samd`(1M) `config` command on the metadata server host:

```
# samd config
```

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

10. (Optional) Create the local hosts configuration file on the new client host.
    You might want to perform this step if your Sun QFS shared host systems have multiple host interfaces. The local hosts configuration file defines the host interfaces that the metadata server and the client hosts can use when accessing the file system. You use this file to specify how file system traffic should flow over public and private networks in your environment.
    For information on creating the local hosts file, see Creating the Local Hosts Configuration File.
11. Issue the `samd`(1M) `config` command on the client host:

```
# samd config
```

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

12. Verify that the `sam-sharefsd` daemon is running for this file system.
    To accomplish this, use the `ps`(1) and `grep`(1) commands as shown in the following code example.

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

    The code example above shows that the `sam-sharefsd` daemon is active for the `sharefs1` file system. If the output returned on your system does not show that the `sam-sharefsd` daemon is active for your Sun QFS shared file system, perform the diagnostic procedures described in the SAM-QFS Troubleshooting Guide.

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

```
# mkdir /sharefs1
```

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

```
# chmod 755 /sharefs1
```

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

15. Modify the `/etc/vfstab` file.
    You must have an entry in the `/etc/vfstab` file for the Sun QFS shared file system. Specify `shared` in the Mount Parameters field. In addition, do one of the following:
    - If you do not want to mount this file system automatically at boot time, type no in the `Mt@boot` field.
    - If you do want the Sun QFS shared file system to automatically mount at boot, do the following:
        - Type yes in the `Mt@boot` field.
        - Add the bg mount option in the `Mt params` field. The bg mount option mounts the file system in the background if

the metadata server is not responding.
The following code example shows the shared and bg entries in the Mt params field.

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

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

```
# df -k
```

The file system should be included in the displayed list.

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

```
# mount /sharefs1
```

For more information about mounting Sun QFS shared file systems, see Mount Options in a Shared File System, or see the mount_samfs(1M) man page.

How to Remove a Client Host From a Shared File System

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

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

2. Use the umount(1M) command to unmount the shared file system on each client host on which the shared file system is mounted.
For example:

```
client# umount sharefs1
```

3. Use the umount(1M) command to unmount the shared file system on the metadata server.
For example:

```
metaserver# umount sharefs1
```

4. If you have not already done so, log in as superuser to the metadata server for the shared file system.
5. Use the samsharefs(1M) command to obtain the current configuration information.
The following example command writes current configuration information to file /etc/opt/SUNWsamfs/hosts.sharefs1:

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

6. Use vi(1) or another editor to open the shared file system information file.
The following code example shows the file before the client host is deleted.

```
# vi /etc/opt/SUNWsamfs/hosts.sharefs1
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host     Host IP          Server      Not     Server
# Name     Addresses        Priority    Used    Host
# ----     ------------     --------    ----    -----
titan      172.16.0.129       1          -      server
tethys     172.16.0.130       2          -
mimas      mimas              -          -
dione      dione              -          -
helene     helene             -          -
```

7. Use the editor to delete the client host or hosts that are no longer to be supported.
   The following code example shows the file after the line for `helene` has been deleted.

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host     Host IP           Server      Not     Server
# Name     Addresses         Priority    Used    Host
# ----     --------------    --------    ----    -----
titan      172.16.0.129        1          -    server
tethys     172.16.0.130        2          -
mimas      mimas               -          -
dione      dione               -          -
```

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

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

   The host `helene` has been removed.

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

```
# samsharefs -R sharefs1
```

10. Use the `mount`(1M) command to mount the shared file system, first on the metadata server and then on each client host in the file system.
    For information about the `mount`(1M) command, see the `mount_samfs`(1M) man page.

Updating the mcf file in a Shared File System Environment

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

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

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

Example – **`samfsconfig`**(1M) Command Example on **`tethys`**

The following example shows how the `samfsconfig`(1M) command is used to retrieve device information for family set `sharefs1` on client `tethys`. Because `tethys` is a potential metadata server, it is connected to the same metadata disks as `titan`, another metadata server in the shared file system.

```
tethys# samfsconfig /dev/dsk/*
#
# Family Set &rdquo;sharefs1&rsquo; Created Wed Jun 27 19:33:50 2003
#
sharefs1                         10 ma sharefs1 on shared
/dev/dsk/c2t50020F23000065EEd0s6 11 mm sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12 mr sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13 mr sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14 mr sharefs1 on
```

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

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

The next example shows the resulting `mcf` file.

Example – **`mcf`** File for **`sharefs1`** Client Host **`tethys`**

```
# Equipment                        Eq  Eq   Family   Dev   Add
# Identifier                       Ord Type Set      State Params
# ----------                       --- ---- ------   ----- ------
sharefs1                           10  ma   sharefs1 on    shared
/dev/dsk/c2t50020F23000065EEd0s6 11  mm   sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12  mr   sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13  mr   sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14  mr   sharefs1 on
```

Example – **`samfsconfig`**(1M) Command Example on **`mimas`**

The following example shows how the `samfsconfig`(1M) command is used to retrieve device information for family set `sharefs1` on client host `mimas`. In this example, `mimas` can never become a metadata server, and it is not connected to the metadata disks.

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

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

- Create a file system Family Set declaration line, beginning with `sharefs1`, in the `mcf` file for client host `mimas`. Enter the `shared` keyword in the Additional Parameters field of the file system Family Set declaration line.
- Create one or more `nodev` lines for each missing Equipment Number entry. For each of these lines, the keyword `nodev` must appear in the Equipment Identifier field for the inaccessible device.
- Ensure that each Device State field is set to on.
- Uncomment the device lines.

The next example shows the resulting `mcf` file for `mimas`.

Example – **`mcf`** File for Client Host **`mimas`**

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

Creating the Local Hosts Configuration File

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

```
/etc/opt/SUNWsamfs/hosts._family-set-name_.local
```

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

The following table shows the fields in the local hosts configuration file.

Table – Local Hosts Configuration File Fields

Field	Content
Host Name	This field must contain the alphanumeric name of a metadata server or potential metadata server that is part of the Sun QFS shared file system.
Host Interfaces	This field must contain a comma-separated list of host interface addresses. This field can be created from the output received from the `ifconfig`(1M) `-a` command. The individual interfaces can be specified in one of the following ways:* Dotted-decimal IP address form* IP version 6 hexadecimal address form* As a symbolic name that the local domain name service (DNS) can resolve to a particular host interface Each host uses this field to determine whether it will try to connect to the specified host interface. The system evaluates the addresses from left to right, and the connection is made using the first responding address in the list that is also included in the shared hosts file.

In a shared file system, each client host obtains the list of metadata server IP addresses from the metadata server host.

The metadata server and the client hosts use both the `/etc/opt/SUNWsamfs/hosts.`fsname file on the metadata server and the `hosts.`fsname`.local` file on each client host (if it exists) to determine the host interface to use when accessing the file system. This process is as follows:

Note -
Client, as in network client, is used to refer to both client hosts and the metadata server host.

The client obtains the list of metadata server host IP interfaces from the file system's on-disk host file.
To examine this file, issue the `samsharefs`(1M) command from the metadata server or from a potential metadata server.
The client searches its files for a `hosts.`fsname`.local` file.
Depending on the outcome of the search, one of the following courses of action is taken:

-
    - If a `hosts.`fsname`.local` file does not exist, the client attempts to connect, in turn, to each address in the system hosts configuration file until it succeeds.
    - If the `hosts.`fsname`.local` file exists, the client performs the following:
      It compares the list of addresses for the metadata server from both the `/etc/opt/SUNWsamfs/hosts.`fsname file on the metadata server and the `hosts.`fsname`.local` file.
      It builds a list of addresses that are present in both places, and then it attempts to connect to each of these addresses, in turn, until it succeeds. If the order of the addresses differs in these files, the client uses the ordering in the `hosts.`fsname`.local` file.

Example – Sun QFS Shared File System Hosts File Example

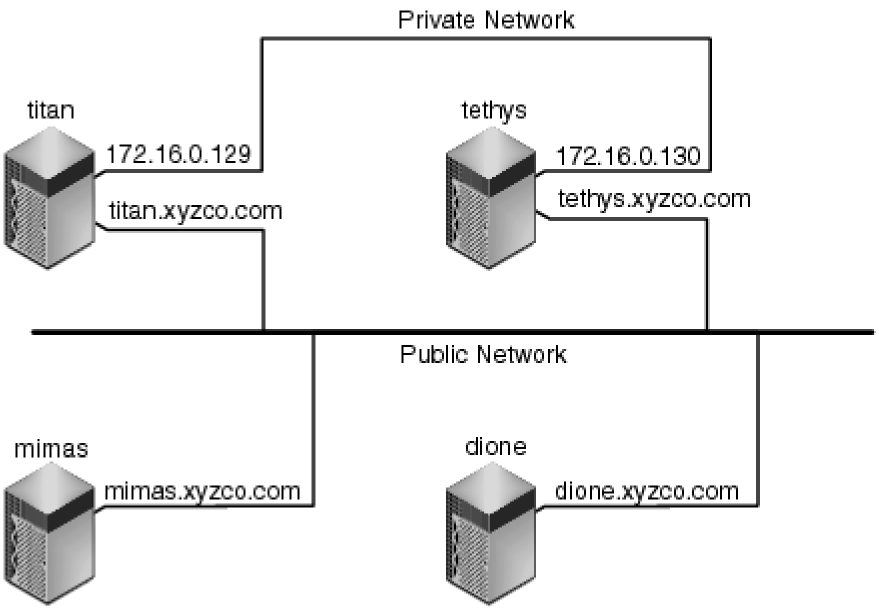
This set of examples shows a detailed scenario for a shared file system that comprises four hosts.

The following example shows a hosts file that lists four hosts.

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host  Host IP           Server     Not    Server
# Name  Addresses         Priority   Used   Host
# ----  ----------------- --------   ----   -----
titan   172.16.0.129        1          -    server
tethys  172.16.0.130        2          -
mimas   mimas               -          -
dione   dione               -          -
```

The following figure shows the interfaces to these systems.

Figure – Network Interfaces for Sun QFS Shared File System Hosts File Example



Example – File **hosts.sharefs1.local** on **titan** and **tethys**

Systems `titan` and `tethys` share a private network connection with interfaces `172.16.0.129` and `172.16.0.130`. To guarantee that `titan` and `tethys` always communicate over their private network connection, the system administrator has created identical copies of `/etc/opt/SUNWsamfs/hosts.sharefs1.local` on each system.

The following example shows the information in the `hosts.sharefs1.local` files on `titan` and `tethys`.

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

Example – File **hosts.sharefs1.local** on **mimas** and **dione**

Systems `mimas` and `dione` are not on the private network. To guarantee that they always connect to `titan` and `tethys` through `titan`'s and `tethys`'s public interfaces, the system administrator has created identical copies of `/etc/opt/SUNWsamfs/hosts.sharefs1.local` on `mimas` and `dione`.

The following example shows the information in the `hosts.sharefs1.local` files on `mimas` and `dione`.

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

Changing the Metadata Server in a Shared File System Environment

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

You can change the metadata server system manually under the following circumstances:

- If the metadata server becomes unavailable
- If you want to change the metadata server or the potential metadata servers

For a metadata server change to succeed, the mount options of the existing metadata server and all potential metadata servers must be the same.

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

- How to Change the Metadata Server When the Metadata Server Is Available
- How to Change the Metadata Server When the Metadata Server Is Unavailable

How to Change the Metadata Server When the Metadata Server Is Available

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

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

   > Note -
   > In archiving environments, you should stop all archiving operations on the metadata server before you issue this command.

How to Change the Metadata Server When the Metadata Server Is Not Available

If the metadata server of a shared file system crashes, it is safe to change the metadata server only after rebooting the metadata server or otherwise ensuring that the server cannot issue any I/O before being rebooted. Do not use any of the following methods to stop the server, because these are likely to corrupt the file system:

- Issuing an L1-A key sequence
- Performing an involuntary failover to another host
- Issuing a `go` (continue) command, requesting a dump file, or issuing a `sync` command to the old metadata server
  Similarly, if the metadata server panics and drops into kernel `adb`(1), do not change the metadata server and then issue a `:c` (continue) command on the server. This action causes the old metadata server to push stale buffers out to the now-active file system.

Use the following steps to change the metadata server:

Steps

1. Ensure that the existing metadata server cannot restart without being rebooted.
   Specifically, ensure that the server is powered down, rebooted, halted, or disconnected from the metadata disks. Your goal is to bring down the old metadata server and flush or destroy all buffers (or otherwise ensure that they cannot be rewritten).
   The following code example shows the key sequence to use from the `kadb` prompt.

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

The following code example shows the key sequence to use from the PROM prompt.

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

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

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

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

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

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

3. (Optional) Unmount the file system.
   Perform this step only if you want to perform a file system check.
   Use the procedure in Unmounting a File System.
4. (Optional) Issue the `samfsck(1M)` command to perform a file system check.
   If the metadata server of a Sun QFS shared file system crashes, the server should be rebooted and the file system should be unmounted on all clients before `samfsck(1M)` is run. The server and clients preallocate blocks before changing the length of files. The `samfsck` (1M) command cleans up files that have extra blocks allocated, and these extra blocks might contain data. If such a cleaned-up file is awaiting a size update from the client, the file will be missing those blocks when the client continues. As a result, the file will be missing data, and the missed data will read as zeroes.

Changing the Metadata Server in an Archiving Environment

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

You can change the metadata server system manually under the following circumstances:

- If the metadata server becomes unavailable
- If you want to change the metadata server or the potential metadata servers

For a metadata server change to succeed, the mount options of the existing metadata server and all potential metadata servers must be the same.

How to Change the Metadata Server in an Archiving Environment

Archiving functions can only run on one host at any time. This procedure assumes that both systems are up at the time of the transfer. In this example, we move archiving functions from host A to host B.

Before carrying out this procedure, verify that host B has access to the robot catalog from host A. The `archiver.cmd` file, `mcf` file, stager.cmd file, and other configuration files must be identical to those on host A.

Steps

1. Idle archiving processes on host A by carrying out the following steps:
   a. Run samcmd aridle and samcmd stidle to halt archiving and staging on host A.
      These commands will allow current archiving and staging to complete, but will not start any new work.
   b. Idle all of the tape drives on host A.
      This can be done with samcmd eq idle, where eq is the equipment number of the drive. This will put the drives in an “off” state after any current I/O completes.
   c. When the archiver and stager are idle and the tape drives are all in the “off” state, run the samd stop command to halt all of the robot and tape-related daemons.
   d. If you have a cron job that runs the recycler, remove this entry from the crontab and verify that the recycler is not currently running.
      At this point, the archiving processes have been halted and file system failover to host B can be performed.
2. Start the archiving processes on host B by running samd config on host B.
   This causes sam-fsd and its subprocesses (archiver, stager, and so on) to reconfigure and re-read the configuration files. It also causes sam-amld and the tape-library-related daemons to start. At this point all Sun QFS shared client applications waiting for stages must reissue the stage requests.
   Host B should now be fully functioning as the archiving process server and metadata server for all file systems.

Converting an Unshared File System to a Shared File System

To perform initial installation and configuration for a shared file system, follow the instructions in Installing Sun QFS. Many examples in this section use host names and configuration information that were introduced in that document.

To convert an unshared file system to a shared file system consists of two tasks:

- First, convert the metadata server.
- Second, add each client to the metadata server. This section describes these procedures.

How to Convert an Unshared Metadata Server to a Shared Metadata Server

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

1. As superuser, log in to the system to be used as the primary metadata server.
2. (Optional) Back up all site-customized system files and configuration files.
   Depending on your software, these files might include mcf, archiver.cmd, defaults.conf, samfs.cmd, or inquiry.conf. Back up these files for all file systems. Also make sure that you have backup copies of files in the /etc/opt/SUNWsamfs directory, and files in the /var/opt/SUNWsamfs directory.
3. Ensure that each file system to be modified is backed up.
   File systems should be backed up regularly according to your site's policies. If you are comfortable with the backup files that already exist for your file systems, there is no need to back them up again now.
4. Use the umount(1M) command to unmount the file system.
   For instructions, see Unmounting a File System.
5. Use the samfsck(1M) -S -F family-set-name command to convert the file system to a Sun QFS shared file system.
   For family-set-name, specify the family set name of the file system that you are converting to a new shared file system. For example:

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

6. Edit the /etc/opt/SUNWsamfs/mcf file to add the shared keyword in the file system’s Additional Parameters field.
   For example:

```
# Equipment Eq Eq Family Dev Add
# Identifier Ord Type Set State Params
# ---------- --- ---- ------ ----- ------
sharefs1 10 ma sharefs1 on shared
/dev/dsk/c2t50020F23000065EEd0s6 11 mm sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12 mr sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13 mr sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14 mr sharefs1 on
```

7. Edit the /etc/vfstab file to add the shared keyword in the file system's Mount Parameters field.
   For example:

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

8. Create the /etc/opt/SUNWsamfs/hosts.fsname hosts configuration file.
   For example:

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host     Host      IP               Server    Not  Server
# Name       Addresses                Priority  Used Host
# ---- ------------------------------ -------- ---- -----
titan   titan-ge                       0 1 - server
tethys  tethys-ge                      0 2 - server
```

   See Creating the Shared Hosts File for more information about creating the hosts configuration file.

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

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

   Note -
   If you see an error message from this command, you can probably ignore the message.

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

```
# samd config
```

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

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

How to Add a Client to the Metadata Server

1. Use the mkdir(1) command to create the mount point for the file system.
   For example:

```
# mkdir /sharefs1
```

2. (Optional) Create an /etc/opt/SUNWsamfs/hosts.file-system-name.local local hosts configuration file.
   You might want to perform this step if your Sun QFS shared host systems have multiple host interfaces. The local hosts configuration file defines the host interfaces that the metadata server and the client hosts can use when accessing the file system. You use this file to specify how file system traffic should flow over public and private networks in your environment.
   The following code example shows a sample local hosts configuration file.

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

For more information on creating the local hosts file, see Creating the Local Hosts Configuration File.

3. If you want to move files from an existing Sun QFS file system into a new Sun QFS shared file system, ensure that each file system to be modified is backed up.
File systems should be backed up regularly according to your site's policies. If you are comfortable with the backup files that already exist for your file systems, there is no need to back them up again now.
4. Use the `umount`(1M) command to unmount the file system.
For instructions, see Unmounting a File System.
5. Edit the `/etc/vfstab` file to add the `shared` keyword in the file system's Mount Parameters field.
For example:

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

6. Create the `/etc/opt/SUNWsamfs/hosts.`fsname hosts configuration file.
The following code example shows a sample.

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IP Server Not Server
# Name Addresses Priority Used Host
# ---- --------- -------- ---- -----
titan titan-ge0 1 - server
tethys tethys-ge0 2 - server
```

For more information about creating the hosts configuration file, see Creating the Shared Hosts File.

Converting a Shared File System to an Unshared File System

To convert a Sun QFS shared file system to an unshared Sun QFS file system requires two tasks:

- Remove the shared clients.
- Convert the metadata server.

This section describes these procedures.

How to Remove a Client From a Shared File System

1. Use the `umount`(1M) command to unmount the file system.
For instructions, see Unmounting a File System.
2. Delete the file system's entry from the `/etc/opt/SUNWsamfs/mcf` file.
3. Delete the file system's entry from the `/etc/vfstab` file.
4. Run the `samd`(1M) `config` command:

```
# samd config
```

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

5. Delete the mount point for the file system.

How to Convert a Shared Metadata Server to an Unshared System

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

1. As superuser, log in to the metadata server system.
2. Back up all site-customized system files and configuration files.

Depending on your software, these files might include mcf(4), archiver.cmd, defaults.conf, samfs.cmd, inquiry.conf, and so on. Back up these files for all file systems. Also make sure that you have backup copies of files in the /etc/opt/SUNWsamfs directory and files in the /var/opt/SUNWsamfs directory.

3. If you want to move files from an existing Sun QFS shared file system into a new Sun QFS file system, ensure that each file system to be modified is backed up.
   File systems should be backed up regularly according to your site's policies. This is described as the last step in the installation procedure. If you are comfortable with the backup files that already exist for your file systems, there is no need to back them up again now.
4. Use the umount(1M) command to unmount the file system.
   For instructions, see Unmounting a File System.
5. Run the samfsck(1M) -F -U file-system-name to convert the Sun QFS shared file system to an unshared file system.
   For file-system-name, specify the name of the Sun QFS shared file system that you are converting to a new unshared file system. For example:

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

6. Edit the /etc/opt/SUNWsamfs/mcf file to remove the shared keyword from the file system's Additional Parameters field.
   For example:

```
# Equipment Eq Eq Family Dev Add
# Identifier Ord Type Set State Params
# ---------- --- ---- ------ ----- ------
samfs1 10 ma samfs1 on
/dev/dsk/c2t50020F23000065EEd0s6 11 mm samfs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12 mr samfs1 on
/dev/dsk/c7t50020F2300006099d0s6 13 mr samfs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14 mr samfs1 on
```

7. Edit the /etc/vfstab file to remove the shared keyword from the file system's Mount Parameters field.
   For example:

```
# File /etc/vfstab
# FS name FS to fsck Mnt pt FS type fsck pass Mt@boot Mt params
samfs1 - /samfs1 samfs - no
```

8. Delete the /etc/opt/SUNWsamfs/hosts.file-system-name configuration file.
9. Run the samd(1M) config command:

```
# samd config
```

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

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

Client-Server Communications in a Shared File System

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

The system generates the following messages to describe status conditions:

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

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

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

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

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

When the shared file system mounts on the server, it generates the following message:

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

Because the metadata server looks up file names on behalf of all clients, performance can be slow with the default size of the Solaris directory name lookup cache (DNLC) on the metadata server. To increase performance when clients are frequently opening a large number of files, you might want to double or even triple the size of this cache from its default.

This procedure is documented in the Solaris Tunable Parameters Reference Manual. The parameter that controls the size of the directory name lookup cache is `ncsize`.

For more information about troubleshooting, see SAM-QFS Troubleshooting.

Configuring the File System

Contents

Configuring the File System

This section provides information about configuring an archiving or standalone file system. For information about configuring a shared file system, see Configuring a Shared File System.

Function of the **`mcf`** File

The master configuration file (`mcf`), located in `/etc/opt/SUNWsamfs/mcf`, describes all devices that the Sun QFS or SAM-QFS software controls or uses. When you create this ASCII file at system configuration time, you declare attributes for each device, and you group the devices in each file system into family sets.

The `mcf` file contains the information that the file systems need in order to identify and organize RAID and disk devices into file systems. It also contains entries for each automated library or device included in a file system. A sample `mcf` file is located in `/opt/SUNWsamfs/examples/mcf`.

For more information about the `mcf` file, see About the Master Configuration File and the `mcf`(4) man page.

Initializing a File System

To create a new file system or to replace an old or damaged file system, use the `sammkfs`(1M) command to initialize the file system. The `sammkfs`(1M) command constructs and initializes file systems.

> **Note -**
> Beginning with Sun QFS 5.0, the `sammkfs` command creates a version 2A file system that has new features, but is not compatible with previous releases. Use the `sammkfs -P` format to create a version 2 file system that is backwards compatible with previous releases. Use the `sammkfs -a` allocation-unit option to specify the DAU setting.
>
> The following `samfsinfo`(1M) command output shows that the `samfs1` file system is using a version 2 superblock.
>
> ```
> # samfsinfo samfs1
> name:     samfs1       version:       2
> time:     Wed Feb 21 13:32:18 1996
> count:    1
> capacity:      001240a0          DAU:         16
> space:         000d8ea0
> ord  eq   capacity      space   device
>   0  10   001240a0   000d8ea0   /dev/dsk/c1t1d0s0
> ```
>
> For more information about features that require a version 2A superblock, or about using the `sammkfs`(1M) command to create the version 2 superblock, see Support for Version 2A Superblock.

The following example shows the `sammkfs`(1M) command in its simplest form, with the file system name as its only argument. This command builds a version 2A superblock for a stand-alone Sun QFS or SAM-QFS file system.

```
# sammkfs samqfs1
```

For more information about the `sammkfs`(1M) command, its options, and the implications of the version 2 and 2A superblocks, see the `sammkfs`(1M) man page . For information about using the `sammkfs`(1M) command to initialize a shared Sun QFS file system, see Configuring a Shared File System.

Configuration Examples

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

Note that all sample configurations could have automated libraries and other removable media devices defined as well, essentially extending the file system beyond the size of the disk cache. Removable media device configurations are shown in only one example. For information about configuring removable media devices, see Configuring Storage Devices for Archiving.

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

How to Create a Round-Robin Disk Configuration

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

This procedure assumes the following:

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

Steps

1. Use an editor to create the `mcf` file, as shown in the following code example.

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

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

```
# mkdir /qfs
```

3. Use the `sammkfs`(1M) command to initialize the file system.
   The following example uses the default 64-kilobyte DAU:

```
# sammkfs qfs1
```

4. Use an editor to modify the `/etc/vfstab` file.
   The Sun QFS file system with `mr` data devices uses striped allocation as a default, so you must set `stripe=0` for round-robin allocation. To explicitly set round-robin on the file system, set `stripe=0`, as follows:

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

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

```
# mount /qfs
```

How to Create a Striped Disk Configuration

In this sample configuration, file data is striped to four data partitions by default.

This procedure assumes the following:

- The metadata device is a single partition (`s6`) used on controller 0, LUN 1. Metadata is written to equipment number 11 only.
- The data devices consist of four disks attached to four controllers. Each disk is on a separate controller.

Steps

1. Use an editor to create the `mcf` file, as shown in the following code example.

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

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

```
# mkdir /qfs
```

3. Use the sammkfs(1M) command to initialize the file system.
The default DAU is 64 kilobytes, but the following example sets the DAU size to 128 kilobytes:

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

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

4. Use an editor to modify the /etc/vfstab file.
The Sun QFS file system uses striped allocation by default. This example sets the stripe width as stripe=1, which is the default. The following setting stripes data across all four of the mr devices with a stripe width of 1 DAU:

```
qfs1    -    /qfs    samfs    -    yes    stripe=1
```

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

```
# mount /qfs
```

How to Create a Striped Group Configuration

Striped groups enable you to group RAID devices together for very large files. A DAU is represented by one bit in the bitmap. If the striped group has n devices, n multiplied by the DAU is the minimum allocation. Only one bit in the bitmap is used to represent n x DAU.

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

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

> Caution -
> Writing a single byte of data fills the entire minimum disk space allocated in a striped group. Striped groups are used for very specific applications. Make sure that you understand the effects of using striped groups with your file system.

Files with lengths less than the aggregate stripe width times the number of devices (in this example, files less than 128 kilobytes x 4 disks = 512 kilobytes in length) still use 512 kilobytes of disk space. Files larger than 512 kilobytes have space allocated for them as needed in total space increments of 512 kilobytes.

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

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

This procedure assumes the following:

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

Steps

1. Use an editor to create the mcf file, as shown in the following code example.

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

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

```
# mkdir /qfs
```

3. Use the `sammkfs`(1M) command to initialize the file system.
   The following example sets the DAU size to 128 kilobytes:

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

4. Use an editor to modify the `/etc/vfstab` file.
   The following example uses the default setting of `stripe=0`, which essentially specifies a round-robin allocation from striped group `g0` to striped group `g1`:

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

This `/etc/vfstab` file sets the stripe width using the `stripe=` option. In this example, there are two striped groups, `g0` and `g1`. With the `stripe=0` specification, files are written round-robin around the two striped groups.

> Note -
> To change the configuration of the striped group after it is created, you must issue another `sammkfs`(1M) command.

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

```
# mount /qfs
```

Configuring WORM-FS File Systems

Contents

Configuring WORM-FS File Systems

About WORM-FS File Systems

Write once read many (WORM) technology is used in many applications for data integrity reasons and because of the accepted legal admissibility of stored files that use the technology.

> Note -
> The WORM-FS package (`SUNWsamfswm`) is included with the Sun QFS software packages, but must be installed separately by using the `pkgadd` command.

The WORM-FS feature offers default and customizable file-retention periods, data and path immutability, and subdirectory inheritance of the WORM setting.

WORM-FS can operate in one of two modes:

- Sun standard compliance mode (referred to as standard mode), which is the default
- Sun emulation compliance mode (referred to as emulation mode), which is designed to provide compatibility with the emulation mode of the Sun StorageTek 5320 network attached storage (NAS) appliance and is similar to an interface defined by Network Appliance

One difference between standard and emulation mode is a restriction on the nature of files that can be retained. Specifically, in standard mode, files with any UNIX executable permissions cannot be retained. There is no such restriction in emulation mode. The restriction in standard mode exists because of the nature of the retention trigger defined for NFS and FTP. For these protocols, retention is requested by specifying that the `setuid` mode be set on the file. Once a file is retained, a client will see the `setuid` mode bit set, but the restriction on executable files will prevent the possible security hole of allowing an executable file owned by the `root` user to be made WORM and therefore impossible to remove. A benefit of this approach is that the user or application can more easily determine which files on the system are indeed WORM-protected files.

Using WORM-FS With NFS Clients

If you are using WORM-FS on Solaris 10 or later with NFS clients connected to it, ensure that NFS version 4 is enabled on the NFS clients and sever.

If you are running an older version of the Solaris OS (before Solaris 10) and NFS version 3, the NFS client might not show the WORM-FS files. In this case, follow these steps:

1. Add the following line to the `/etc/system` file:

```
set nfs:nfs_allow_preepoch_time = 1
```

2. Reboot the system.

Enabling the WORM-FS Feature

There are four mount options that can be used to enable the WORM-FS feature:

The following table shows four mount options that you can use to enable the WORM-FS feature.

Option	Brief Description	Notes
`worm_capable`	Standard WORM mode	The WORM trigger command, `chmod 4000` file-name/directory-name, is used to set the WORM bit on a file or directory.
`worm_lite`	Relaxes some of the standard WORM mode restrictions	The system administrator is allowed to delete files before retention expiration and reduce the file retention period. File data and path integrity remain immutable. See WORM "Lite" Options for more information.

`worm_emul`	WORM emulation mode, which is designed to provide compatibility with the emulation mode of the Sun StorageTek 5320 network attached storage (NAS) appliance	This provides standard WORM functionality with a different WORM trigger. The WORM bit is set by changing a directory or file from writeable to read-only.
`emul_lite`	WORM emulation mode, which is designed to provide compatibility with the "lite" version of the Sun StorageTek 5320 network attached storage (NAS) appliance	This provides standard WORM lite functionality with a different WORM trigger. The WORM bit is set by changing a directory or file from writeable to read-only. As with the worm_lite option, the administrator can carry out special operations on files. See WORM "Lite" Options for more information.

These four mount options are somewhat exclusive. You can upgrade from "lite" to standard WORM mode, but you cannot change from standard WORM mode to emulation mode, or from emulation to standard mode. These options can be provided on the command line when the file system is mounted, listed in `/etc/vfstab`, or provided in `/opt/SUNWsamfs/samfs.cmd`. The normal rules of precedence for mount options apply.

The WORM attribute is stored in the mount table and enables WORM files to be created in directories anywhere in the file system.

> Note -
> You must have system administration privileges to set a WORM mount option in `/etc/vfstab`.

The following example shows an example of WORM-FS mount options. The file system `samfs1` mounted at `/samfs1` is WORM-capable and has the default retention period for files set to 60 minutes.

Example – Using WORM-FS Mount Options

```
# cat /etc/vfstab
  #device device mount FS fsck mount mount
  #to mount to fsck point type pass at boot options
  #
  fd - /dev/fd fd - no -
  /proc - /proc proc - no -
  /dev/dsk/c0t0d0s1 - - swap - no -
  samfs1 - /samfs1 samfs - yes worm_capable,def_retention=60
  swap - /tmp tmpfs - yes -
```

After the WORM-FS feature has been enabled and at least one WORM file is resident in the file system, the file system's superblock is updated to reflect the WORM capability. Any subsequent attempt to rebuild the file system through `sammkfs` will fail, unless you are using the `worm_lite` or `emul_lite` mount option.

WORM "Lite" Options

The `worm_lite` and `emul_lite` mount options create a modified WORM environment that eases the restrictions on actions that can be taken on WORM-enabled volumes and retained files. The WORM lite options can be a solution for companies with document management and retention policies requiring data retention guarantees but not the strict constraints that WORM places on systems. Mechanisms exist to alter and even reverse some data retention decisions.

The WORM lite options can also be used for testing and configuring WORM systems and applications before upgrading to the more strict standard WORM policies.

The WORM lite environment behaves similarly to the standard WORM mode. File data and path remain immutable, but the system administrator is allowed to carry out the following special actions:

- Remove WORM files before the retention time has expired
- Shorten the retention time on WORM files
- Delete WORM Lite-enabled volumes or rebuild them using the `sammkfs` command

Creating WORM Files

A WORM mount option enables a file system to contain WORM files, but it does not automatically create WORM files. To create a WORM file, you must first make the directory WORM-capable. To do this, create an ordinary directory and then use a WORM trigger command to set the WORM bit on the directory. Depending on the mount option being used, the following WORM trigger commands are available:

- Use `chmod 4000` directory-name to set the WORM bit if you are using the `worm_capable` or `worm_lite` mount option.
- Remove the write permissions on the directory to set the WORM bit if you are using the `worm_emul` or `emul_lite` mount option.

After setting the WORM bit on a directory, you can create files in that directory and then use the appropriate WORM trigger to set the WORM bit on files that you want retained. The WORM trigger is the same for both files and directories.

The following are examples of using the WORM trigger for each of the four mount options using the system-wide default retention value:

Example 1. WORM trigger is **`chmod 4000`**

Simple application of the WORM trigger using standard WORM functionality:

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -       /samfs1  samfs   -       no     bg,worm_capable

[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod 4000 WORM
[root@ns-east-44]# sls -D

WORM:
  mode: drwxr-xr-x  links:   2  owner: root      group: root
  length:      4096  admin id:      0  inode:     1025.1
  access:      Jan 30 15:50  modification: Jan 30 15:50
  changed:     Jan 30 15:50  attributes:   Jan  1  1970
  creation:    Jan 30 15:50  residence:    Jan 30 15:50
  worm-capable        retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod 4000 test
[root@ns-east-44]# sls -D

test:
  mode: -r-Sr--r--  links:   1  owner: root      group: root
  length:         0  admin id:      0  inode:     1026.3
  access:      Jan 30 15:51  modification: Jan 30 15:51
  changed:     Jan 30 15:51  retention-end: Mar  1 15:51 2007
  creation:    Jan 30 15:51  residence:    Jan 30 15:51
  retention:   active        retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
rm: test not removed: Read-only file system
[root@ns-east-44]# ls
test
```

Example 2. WORM trigger is ***`chmod 4000`***

Simple application of the WORM trigger using standard WORM lite functionality:

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -       /samfs1  samfs   -       no     bg,worm_lite

[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod 4000 WORM
[root@ns-east-44]# sls -D

WORM:
  mode: drwxr-xr-x  links:   2  owner: root      group: root
  length:      4096  admin id:      0  inode:     1025.1
  access:      Jan 30 16:12  modification: Jan 30 16:12
  changed:     Jan 30 16:12  attributes:   Jan  1  1970
  creation:    Jan 30 16:12  residence:    Jan 30 16:12
  worm-capable        retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod 4000 test
[root@ns-east-44]# sls -D
test:
  mode: -r-Sr--r--  links:   1  owner: root      group: root
  length:         0  admin id:      0  inode:     1026.1
  access:      Jan 30 16:13  modification: Jan 30 16:13
  changed:     Jan 30 16:13  retention-end: Mar  1 16:13 2007
  creation:    Jan 30 16:13  residence:    Jan 30 16:13
  retention:   active        retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
[root@ns-east-44]# ls
[root@ns-east-44]#
```

Example 3. WORM trigger is ***chmod -w***

Simple application of the WORM trigger using WORM emulation mode:

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -       /samfs1  samfs   -       no     bg,worm_emul

[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod -w WORM
[root@ns-east-44]# sls -D

WORM:
  mode: drwxr-xr-x  links:   2  owner: root      group: root
  length:      4096  admin id:      0  inode:     1025.1
  access:      Jan 30 16:26  modification: Jan 30 16:26
  changed:     Jan 30 16:26  attributes:   Jan  1  1970
  creation:    Jan 30 16:26  residence:    Jan 30 16:26
  worm-capable        retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod -w test
[root@ns-east-44]# sls -D

test:
  mode: -r--r--r--  links:   1  owner: root      group: root
  length:         0  admin id:      0  inode:     1026.1
  access:      Jan 30 16:26  modification: Jan 30 16:26
  changed:     Jan 30 16:26  retention-end: Mar  1 16:26 2007
  creation:    Jan 30 16:26  residence:    Jan 30 16:26
  retention:   active        retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
rm: test not removed: Read-only file system
[root@ns-east-44]# ls
test
```

Example 4. WORM trigger is ***chmod -w***

Simple application of the WORM trigger using WORM emulation lite mode:

```
[root@ns-east-44]# grep -i worm /etc/vfstab
samfs1 -       /samfs1  samfs   -       no     bg,emul_lite

[root@ns-east-44]# mount samfs1
[root@ns-east-44]# cd /samfs1
[root@ns-east-44]# mkdir WORM
[root@ns-east-44]# chmod -w WORM
[root@ns-east-44]# sls -D

WORM:
  mode: drwxr-xr-x  links:   2  owner: root      group: root
  length:      4096  admin id:      0  inode:     1025.1
  access:      Jan 30 16:36  modification: Jan 30 16:36
  changed:     Jan 30 16:36  attributes:   Jan  1  1970
  creation:    Jan 30 16:36  residence:    Jan 30 16:36
  worm-capable        retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod -w test
[root@ns-east-44]# sls -D

test:
  mode: -r--r--r--  links:   1  owner: root      group: root
  length:         0  admin id:      0  inode:     1026.1
  access:      Jan 30 16:36  modification: Jan 30 16:36
  changed:     Jan 30 16:36  retention-end: Mar  1 16:36 2007
  creation:    Jan 30 16:36  residence:    Jan 30 16:36
  retention:   active        retention-period: 0y, 30d, 0h, 0m

[root@ns-east-44]# rm test
rm: test: override protection 444 (yes/no)? yes
[root@ns-east-44]# ls
[root@ns-east-44]#
```

Note -
Use care when applying the WORM trigger. The file data and path cannot be changed after the file has the WORM feature applied. Once this feature is applied to a file, it is irrevocable. Further, once the WORM trigger is applied to a file, its volume also become a WORM volume and remains that way. The volume can only be destroyed using a volume management or RAID interface. If one of the WORM "lite" options was used to create it, the volume can also be rebuilt by using `sammkfs`.

Retention Periods

The WORM-FS feature also includes file-retention periods that can be customized. Assigning a retention period to a file maintains the WORM features in that file for the specified period of time.

Note -
Retention periods cannot extend beyond 01/18/2038 when initially assigning or extending the period using Solaris/UNIX utilities. This is due to the fact these utilities use signed 32 bit numbers to represent time in seconds. Time is measured from the epoch which is January 1, 1970. 2**31 seconds from the epoch extends to 01/18/2038 around 10:14 PM. You can, however, exceed this date using a default retention period. See Setting the Default Retention Period.

Do one of the following to set a retention period for a file:

- Advance the file's access time using the `touch` utility, or with a program using the `libc` subroutine `utimes()`. The file's retention period is stored in minutes. After the access time is advanced, use the appropriate WORM trigger to set the WORM bit.
- Use the default retention period for a file. This is accomplished by applying the appropriate WORM trigger and allowing the file system to apply the default retention period. See Setting the Default Retention Period for more information.

The following example shows the creation of a file in a WORM-capable directory, using the WORM trigger on the file (with the `chmod 4000` command), and using the `sls` command to display the file's WORM features. This example uses the default retention period of the file system (60 minutes, as set in Example – Using WORM-FS Mount Options).

Example – Creation of a WORM-Capable Directory and WORM File

```
# cd WORM
# echo "This is a test file" >> test
# sls -D
test:
            mode: -rw-r--r-- links: 1 owner: root group: other
            length: 20 admin id: 0 inode: 1027.1
            access: Oct 30 02:50 modification: Oct 30 02:50
            changed: Oct 30 02:50 attributes: Oct 30 02:50
            creation: Oct 30 02:50 residence: Oct 30 02:50

            checksum: gen no_use not_val algo: 0

# chmod 4000 test
# sls -D
test:
            mode: -r--r--r-- links: 1 owner: root group: other
            length: 20 admin id: 0 inode: 1027.1
            access: Oct 30 02:50 modification: Oct 30 02:50
            changed: Oct 30 02:50 retention-end: Oct 30 2005 03:50
            creation: Oct 30 02:50 residence: Oct 30 02:50
            retention: active retention-period: 0y, 0d, 1h, 0m
            checksum: gen no_use not_val algo: 0
```

With the addition of the WORM-FS feature, three states are possible for a file in a Sun QFS file system:

- Normal
- Retained
- Expired

The normal state represents the state of an ordinary file in a Sun QFS file system. A transition to the retained, or active, state occurs when the WORM bit is set on a file. The expired, or over, state occurs when the file's retention period is exceeded.

When a retention period is assigned to a file and the WORM trigger is applied to it, the file's path and data are immutable. When the retention period expires, the state is changed to "expired" but the path and data remain immutable.

When a file is in an expired state, only two operations are available:

- Extension of the retention period (The retention period cannot be shortened unless you are using a WORM "lite" option.)
- Deletion of the file

If the retention period is extended, the file's state returns to "active" and the new end date and duration are set accordingly.

Both hard and soft links to files can be used with the WORM-FS feature. Hard links can be established only with files that reside in a WORM-capable directory. After a hard link is created, it has the same WORM characteristics as the original file. Soft links can also be established, but a soft link cannot use the WORM features. Soft links to WORM files can be created in any directory in a Sun QFS file system.

Another attribute of the WORM-FS feature is directory inheritance. New directories created under a directory that includes a WORM attribute inherit this attribute from their parent. If a directory has a default retention period set, this retention period is also inherited by any new subdirectories. The WORM bit can be set on any file whose parent directory is WORM-capable. Ordinary users can set the WORM feature on directories and files that they own or have access to by using normal UNIX permissions.

> Note -
> A WORM-capable directory can only be deleted if it contains no WORM files.

Setting the Default Retention Period

The default retention period for a file system can be set as a mount option in the `/etc/vfstab` file. For example:

```
samfs1 - /samfs1 samfs - nobg,worm_emul,def_retention=1y60d
```

The format for setting the default retention period is MyNdOhPm, in which M, N, O, and P are non-negative integers and `y`, `d`, `h`, and `m` stand for years, days, hours, and minutes, respectively. Any combination of these units can be used. For example, `1y5d4h3m` indicates 1 year, 5 days, 4 hours, and 3 minutes; `30d8h` indicates 30 days and 8 hours; and `300m` indicates 300 minutes. This format is backward compatible with software versions prior to 4U5, in which the retention period was specified in minutes. It is important to note, although the granularity of the period is in minutes, the accuracy of the period is based on one day. Also, the function handling days, hours, and minutes does not account for leap years when determining retention periods. You must consider this when using one (or all) of these to set the default retention period.

You can also use the default retention period to set a file or directory's retention period beyond the year 2038. To do this, set the default retention period to a value which exceeds 2038 and mount the file system. Then use the appropriate WORM trigger to apply the default retention period. Here is an example of using the default retention period to set a retention period on a directory and file which exceeds the year 2038.

Example – Extending the Retention Period Beyond 2038

```
[root@ns-east-44]# grep samfs1 /etc/vfstab
samfs1 - /samfs1 samfs - no
bg,worm_capable,def_retention=34y
    [root@ns-east-44]# mount samfs1
    [root@ns-east-44]# cd /samfs1
    [root@ns-east-44]# mkdir WORM
    [root@ns-east-44]# chmod 4000 WORM
    [root@ns-east-44]# sls -D
WORM:
        mode: drwxr-xr-x links: 2 owner: root group: root
        length: 4096 admin id: 0 inode: 1026.1
        access: Feb 20 14:24 modification: Feb 20 14:24
        changed: Feb 20 14:24 attributes: Jul 26 1970
        creation: Feb 20 14:24 residence: Feb 20 14:24
        worm-capable retention-period: 34y, 0d, 0h, 0m

[root@ns-east-44]# cd WORM
[root@ns-east-44]# touch test
[root@ns-east-44]# chmod 4000 test
[root@ns-east-44]# sls -D
test:
        mode: -r-Sr--r-- links: 1 owner: root group: root
        length: 0 admin id: 0 inode: 1027.1
        access: Feb 20 14:24 modification: Feb 20 14:25
        changed: Feb 20 14:25 retention-end: Feb 20 14:25 2041
            creation: Feb 20 14:24 residence: Feb 20 14:24
    retention: active retention-period: 34y, 0d, 0h, 0m
```

You can also set a default retention period for a directory using the `touch` utility, as described in the following section, Setting the Retention Period Using `touch`. This retention period overrides the default retention period for the file system and is inherited by any subdirectories.

Setting the Retention Period Using **touch**

You can use the `touch` utility to set or extend a file's or directory's retention period. You can also use `touch` to shorten the default retention period for a directory (but not for a file).

To set the retention period, you must first advance the file's or directory's access time using `touch`, and then apply the WORM trigger by using the `chmod` command or removing write permissions (depending on the WORM mode in place at the time).

The following example shows the use of the `touch` utility to set a file's retention period, followed by the application of the WORM trigger.

Example – Using **touch** and **chmod** to Set the Retention Period

```
# touch -a -t200508181125 test
# sls -D
test:
        mode: -rw-r--r-- links: 1 owner: root group: root
        length: 0 admin id: 0 inode: 1027.1
        access: Aug 18 2005 modification: Aug 18 11:19
        changed: Aug 18 11:19 attributes: Aug 18 11:19
        creation: Aug 18 11:19 residence: Aug 18 11:19

# chmod 4000 test
# sls -D
test:
        mode: -r-Sr--r-- links: 1 owner: root group: root
        length: 0 admin id: 0 inode: 1027.1
        access: Aug 18 2005 modification: Aug 18 11:19
        changed: Aug 18 11:19 retention-end: Aug 18 2005 11:25
        creation: Aug 18 11:19 residence: Aug 18 11:19
        retention: active retention-period: 0y, 0d, 0h, 6m
```

The `-a` option for `touch` is used to change the access time of the file or directory. The `-t` option specifies what time is to be used for the access time field. The format for the time argument is `[[CC]YY]MMDDhhmm[.SS]`, as follows:

- `[CC]` – The first two digits of the year.
- `[YY]` – The second two digits of the year.
- `MM` – The month of the year (01–12).
- `DD` – The day of the month (01–31).
- `hh` – The hour of the day (00–23).
- `mm` – The minute of the hour (00–59).
- `[SS]`– The second of the minute (00–61).

The CC, YY, and SS fields are optional. If CC and YY are not given, the default is the current year. See the `touch` manpage for more information on these options.

To set the retention period to permanent retention, set the access time to its largest possible value: `203801182214.07`.

Extending a File's Retention Period

This example shows an example of using `touch` to extend a file's retention period.

Example – Using **`touch`** to Extend a File's Retention Period

```
# sls -D test
test:
        mode: -r-Sr--r-- links: 1 owner: root group: root
        length: 0 admin id: 0 inode: 1029.1
        access: Aug 18 11:35 modification: Aug 18 11:33
        changed: Aug 18 11:33 retention-end: Aug 18 2005 11:35
        creation: Aug 18 11:33 residence: Aug 18 11:33
        retention: over retention-period: 0y, 0d, 0h, 2m
# touch -a -t200508181159 test
# sls -D
test:
        mode: -r-Sr--r-- links: 1 owner: root group: root
        length: 0 admin id: 0 inode: 1029.1
        access: Aug 18 11:35 modification: Aug 18 11:33
        changed: Aug 18 11:33 retention-end: Aug 18 2005 11:59
        creation: Aug 18 11:33 residence: Aug 18 11:33
        retention: active retention-period: 0y, 0d, 0h, 26m
```

In this example the retention period was extended to Aug 18, 2005 at 11:59AM, which is 26 minutes from the time the WORM trigger was initially applied.

Note -
Using touch to extend the retention period is independent of the active WORM mode.

Using **sls** to View WORM-FS Files

Use the sls command to view WORM file attributes. The -D option shows whether a directory is WORM-capable. Use this option on a file to display when the retention period began, when it will end, the current retention state, and the duration as specified on the command line.

The retention period start time and duration (in minutes) are stored in the file's inode.
To access this information directly one must use a program similar to the following example:

Example – Program for Direct Access to Retention Period Start Time and Duration

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
/*
* SAMFS includes in /opt/SUNWsamfs/include
*/
#include "stat.h"
#include "lib.h"

/*
* SAMFS libraries in /opt/SUNWsamfs/lib
*/
int
main(int argc, char **argv)
{
   char ibuf[1000];
   struct sam_stat buf;

   if (argc != 2) {
       printf("usage: sam_worm_stat filename\n");
       exit(-1);
   }

   if (sam_stat(argv[1], &buf, sizeof(buf)) == 0) {
       (void)time_string(buf.rperiod_start_time,
buf.rperiod_start_time, ibuf);
       printf("retention period start is %s\n", ibuf);
       printf("retent period duration is %ld minutes\n",
buf.rperiod_duration);
   } else {
       printf("can not sam_worm_stat %s\n", argv[1]);
   }
}
```

The following example shows how sls -D displays a file's retention status.

Example – Using **sls** to Find a File's Retention Status

```
# sls -D test
test:
        mode: -r-Sr--r-- links: 1 owner: root group: root
        length: 5 admin id: 0 inode: 1027.1
        access: Aug 18 2008 modification: Aug 18 11:19
        changed: Aug 18 11:19 retention-end: Aug 18 2008 11:25
        creation: Aug 18 11:19 residence: Aug 18 11:19
        retention: active retention-period: 0y, 0d, 0h, 6m
```

In this example, the retention state is active, as shown by the retention: active designation, meaning that the file has the WORM bit set. The retention period started on August 18, 2008, at 11:19 and will end on August 18, 2008, at 11:25. The retention period was specified to be 0 years, 0 days, 0 hours, and 6 minutes.

Using `sfind` to Find WORM-FS Files

Use the `sfind` utility to search for files that have certain retention periods. See the `sfind(1)` man page for more information on the options. The following options are available:

- `-ractive` – Finds files whose retention period is active.
- `-rover` – Finds files whose retention periods have expired.
- `-rafter` date – Finds files whose retention period will end after the specified date. The date is specified as `YYYYMMDDHHmm`, where `YYYY` is the year, `MM` is the month, `DD` is the day, `HH` is the hour, and `mm` is minutes.
  The following example shows the command to find files whose retention period expires after 12/24/2004 at 15:00.

```
# sfind -rafter 200412241500
```

- `-rremain` time – Finds files that have retention periods with at least the specified amount of time left. The time is specified as `MyNdOhPm`, where `M`, `N`, `O`, and `P` are arbitrary non-negative integers and `y`, `d`, `h`, and `m` represent the number of years, days, hours, and minutes, respectively.
  For example, the following command finds files for which more than 1 year, 10 days, 5 hours, and 10 minutes remain before expiration.

```
# sfind -rremain 1y10d5h10m
```

- `-rlonger` time – Finds files that have retention periods longer than the specified amount of time. The time is specified as `MyNdOhPm`, where `M`, `N`, `O`, and `P` are arbitrary non-negative integers and `y`, `d`, `h`, and `m` represent the number of years, days, hours, and minutes, respectively.
  For example, the following command finds files that have retention periods longer than 10 days.

```
# sfind -rlonger 10d
```

- `-rpermanent` – Finds files whose retention period is permanent.

File Allocation Methods

Contents

File Allocation Methods

The Sun QFS software enables you to specify both round-robin and striped allocation methods. The following table shows the default file allocation methods used.

Table – Default Allocation Methods

File System	Metadata	File Data
Standalone file system	Striped	Striped
Shared file system	Striped	Round-robin

Striped groups	Striped	Round-robin

The rest of this section describes allocation in more detail.

Metadata Allocation

Metadata allocation varies according to the type of file system:

- For ms file systems, metadata is allocated across the md devices.
- For ma file systems, metadata is allocated across the mm devices. No file data is allocated on the mm devices.

Inodes are 512 bytes in length. Directories are initially 4 kilobytes in length. The following table shows how the system allocates metadata.

Table – Metadata Allocation

Metadata Type	Allocation Increments for **ma** File Systems	Allocation Increments for **ms** File Systems
Inodes (`.inodes` file)	16-kilobyte DAU	16-, 32-, or 64-kilobyte DAU
Indirect blocks	16-kilobyte DAU	16-, 32-, or 64-kilobyte DAU
Directories	4-kilobyte blocks and 16-kilobyte DAUs	4 kilobytes, up to a 32- kilobyte total, then DAU size

Round-Robin Allocation

The round-robin allocation method writes one data file at a time to each successive device in the family set. Round-robin allocation is useful for multiple data streams, because in this type of environment aggregate performance can exceed striping performance.

Round-robin disk allocation enables a single file to be written to a logical disk. The next file is written to the next logical disk, and so on. When the number of files written equals the number of devices defined in the family set, the file system starts over again with the first device selected. If a file exceeds the size of the physical device, the first portion of the file is written to the first device, and the remainder of the file is written to the next device with available storage. The size of the file being written determines the I/O size.

To specify round-robin allocation explicitly, enter `stripe=0` in the `/etc/vfstab` file.

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

Figure 1 – Round-Robin Allocation in an **ms** File System Using Five Devices

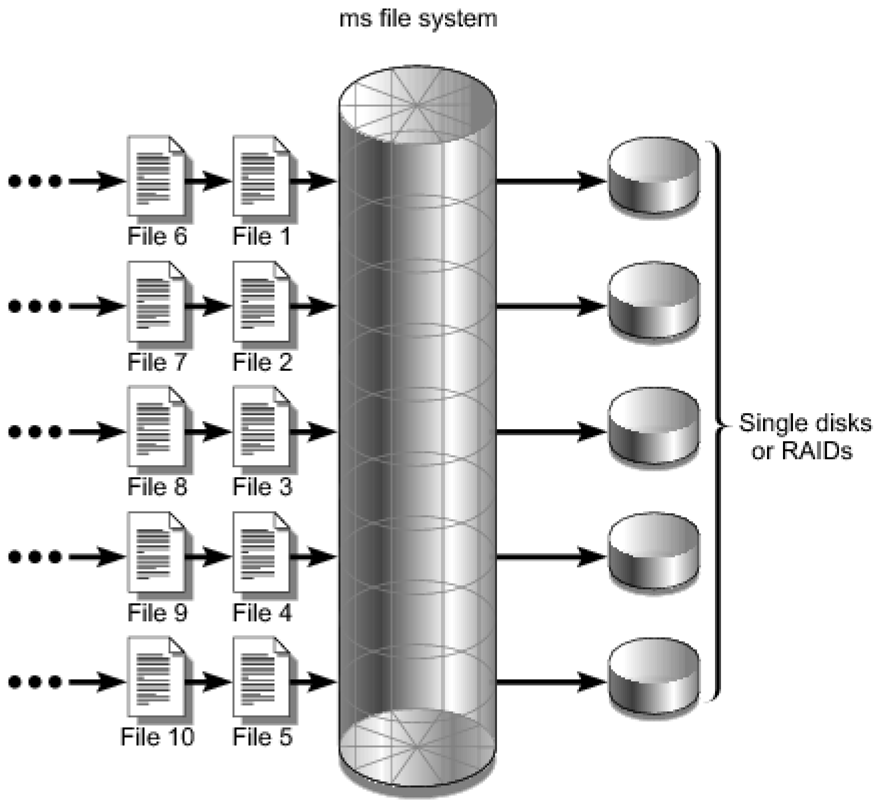
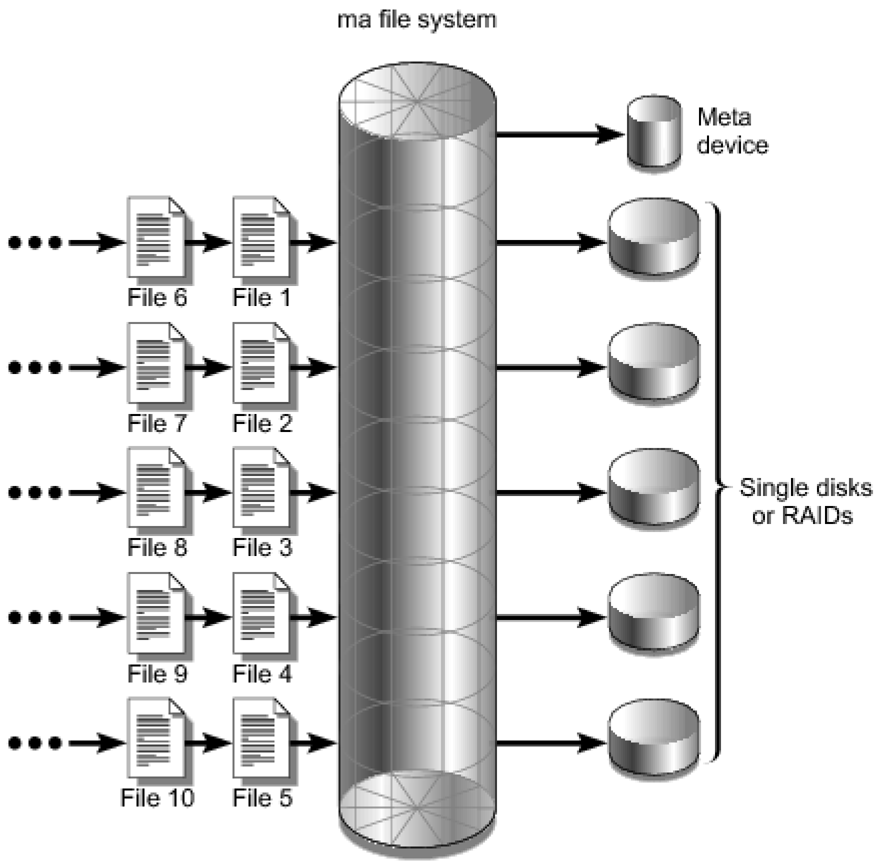


Figure 2 – Round-Robin Allocation in an **ma** File System Using Five Devices



Striped Allocation

By default, Sun QFS file systems use a striped allocation method to spread data over all the devices in the file system family set. Striping is a method of concurrently writing files in an interlaced fashion across multiple devices.

Striping is used when performance for one file requires the additive performance of all the devices. A file system that uses striped devices addresses blocks in an interlaced fashion rather than sequentially. Striping generally increases performance because it enables multiple I/O streams to write a file simultaneously across multiple disks. The DAU and the stripe width determine the size of the I/O transmission.

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

When a file system writes a file to an `md` device, it starts by trying to fit the file into a small DAU, which is 4 kilobytes. If the file does not fit into the first eight small DAUs (32 kilobytes) allocated, the file system writes the remainder of the file into one or more large DAUs.

When a file system writes a file to an `mr` device, it writes first to one DAU, then to another, and so on. The `mr` devices have only one DAU size.

Multiple active files cause significantly more disk head movement with striped allocation than with round-robin allocation. If I/O is to occur to multiple files simultaneously, use round-robin allocation.

The following figures depict `ms` and `ma` file systems that use striped allocations. In these figures, `DAU x stripe-width` bytes of the file are written to disk 1. `DAU x stripe-width` bytes of the file are written to disk 2 and so on. The order of the stripe is first-in-first-out for the files. Striping spreads the I/O load over all the disks.

Figure 3 – Striping in an **ms** File System Using Five Devices

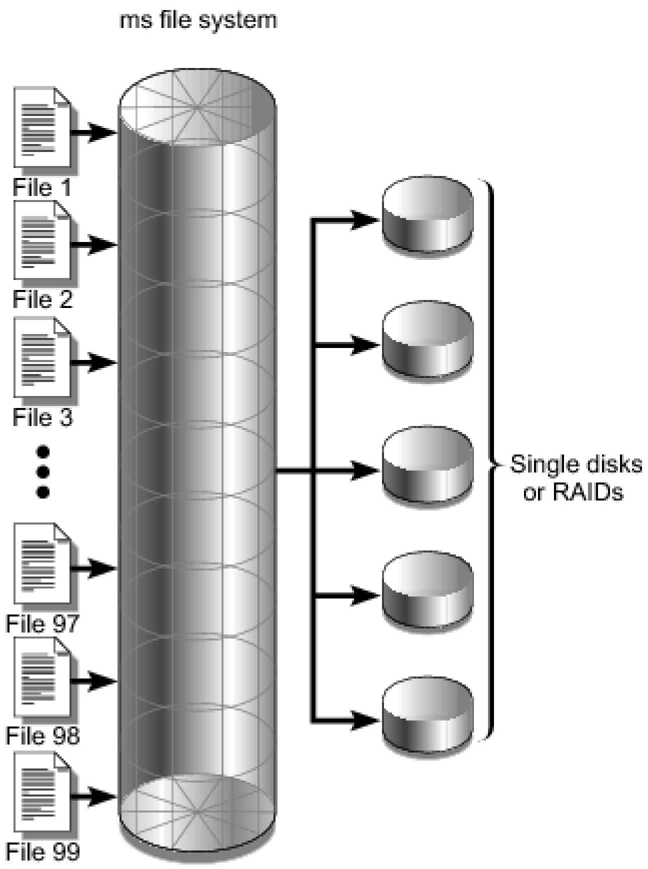
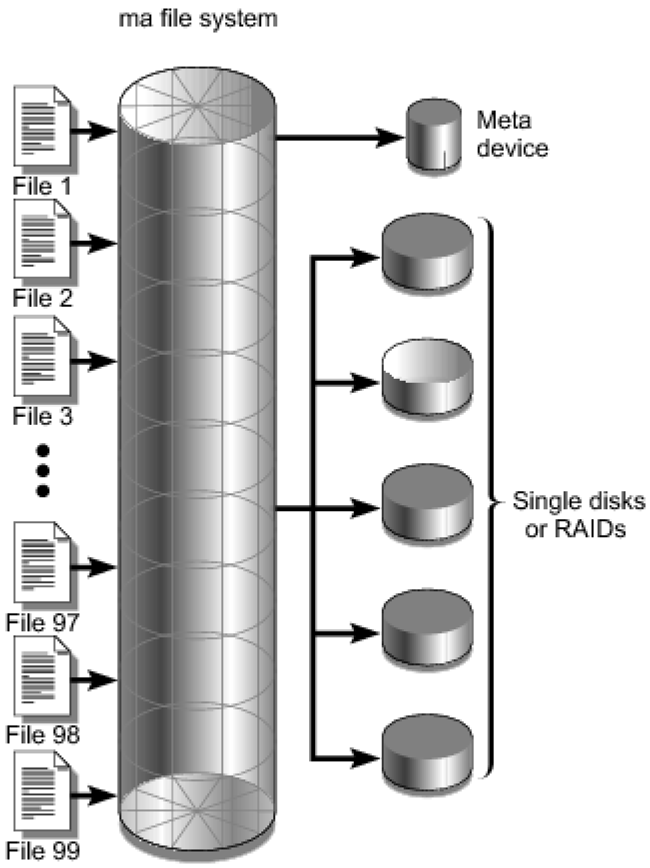


Figure 4 – Striping in an **ma** File System Using Five Devices



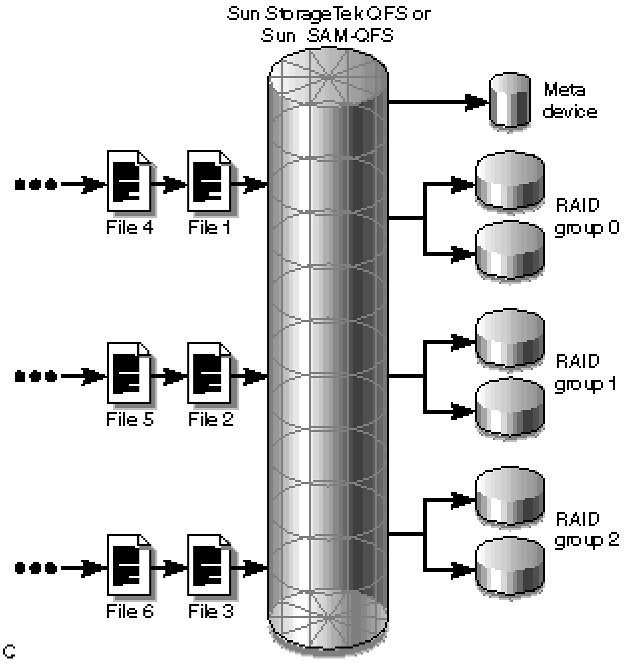
Striped Groups

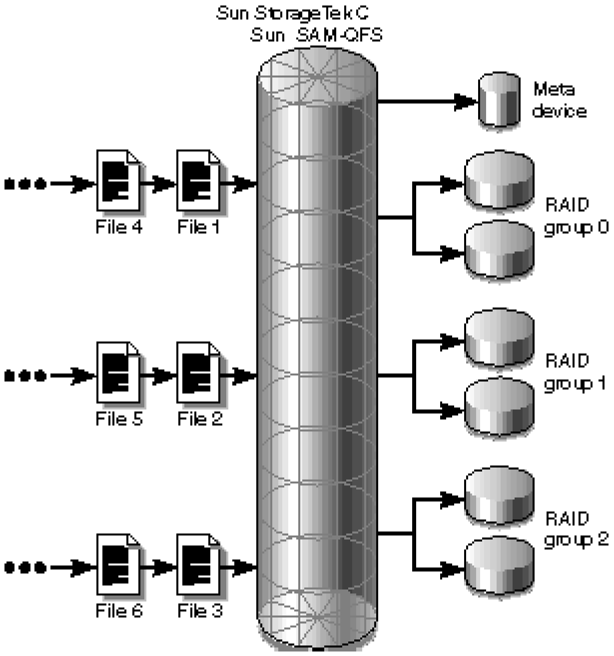
A striped group is a Sun QFS allocation method designed for file systems that have extremely large I/O requirements and terabytes of disk cache. A striped group enables you to designate an Equipment Type value that accounts for multiple physical disks. Multiple striped group Equipment Type entries can make up a single Sun QFS file system. Striped groups save bitmap space and system update time for very large RAID configurations.

A striped group is a collection of devices within a Sun QFS file system. Defined in the `mcf` file as gXXX devices, striped groups enable one file to be written to and read from two or more devices. You can specify up to 128 striped groups within a file system.

Figure 5 – Sun QFS Round-Robin Striped Groups

The following figure depicts an `ma` file system using striped groups and a round-robin allocation. In this figure, files written to the `qfs1` file system are allocated round-robin among the defined striped groups `g0`, `g1`, and `g2`. Each group consists of two physical RAID devices.





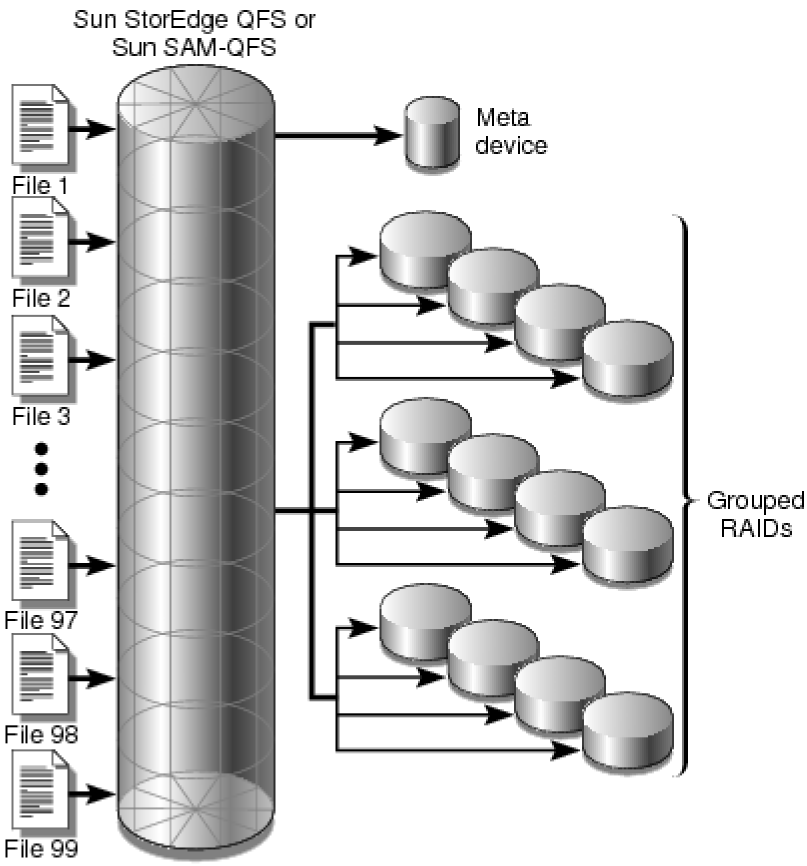
For the configuration in the above figure, the mount point option in `/etc/vfstab` is set to `stripe=0`. The following example shows the `mcf` file that declares these striped groups.

Example – **`mcf`** File Showing Striped Groups

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

Figure 6 – Sun QFS Striped Group Allocation

The following figure depicts a Sun QFS `ma` file system using striped groups and striped allocation. Files written to the `qfs1` file system are striped through groups `g0`, `g1`, and `g2`. Each group includes four physical RAID devices. The mount point option in `/etc/vfstab` is set to `stripe=1` or greater.



Mismatched Striped Groups

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

> Note -
> In a file system that contains mismatched striped groups, a single file cannot span multiple stripe groups. If the stripe group on which the file resides fills, it cannot be extended. If mismatched stripe groups are present, use the `setfa`(1) command's `-g` option to direct files into the desired group. For more information, see the `setfa`(1) man page.
> To determine how full a stripe group is, use the `samu`(1M) operator utility, and access the `m` display to display the status of mass storage.

The following example shows how a file system can be set up with mismatched striped groups to store different types of files.

Example of a Mismatched Striped Group

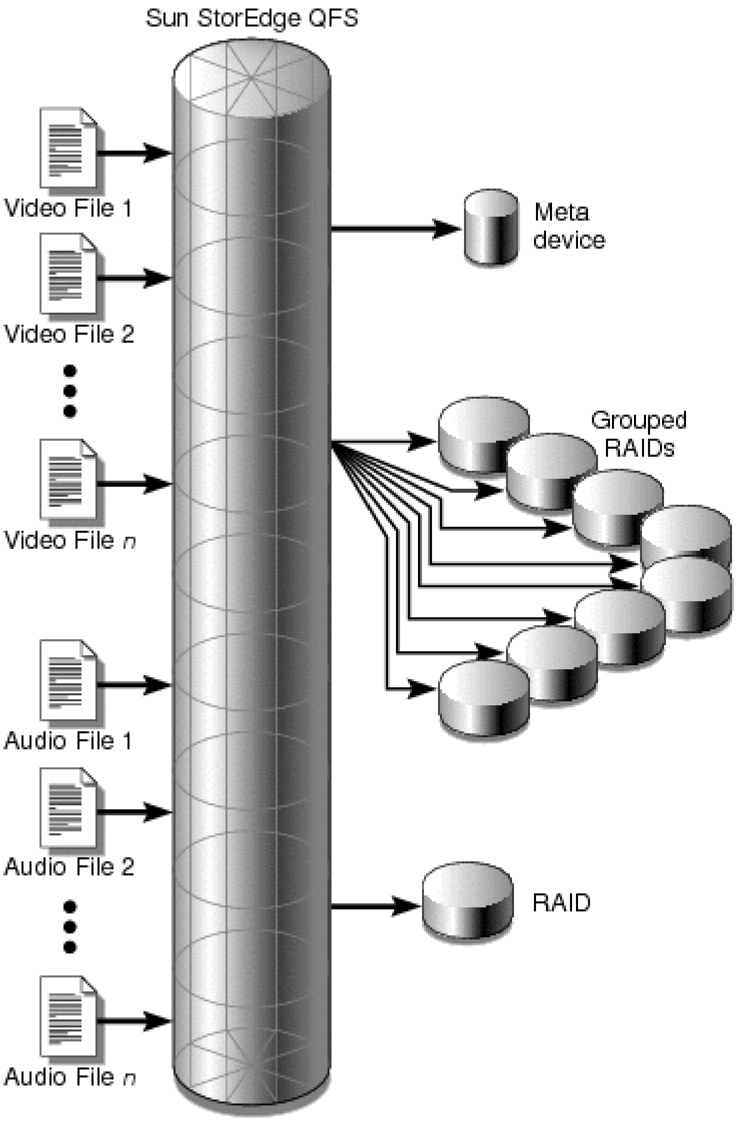
Suppose that you need to create a file system at your site that contains both video and audio data.

Video files are quite large and require greater performance than audio files. You want to store them in a file system with a large striped group, because striped groups maximize performance for very large files.

Audio files are smaller and require lower performance than video files. You want to store them in a small striped group. One file system can support both video and audio files.

The following figure depicts the file system needed. It is an `ma` file system that uses mismatched striped groups in a striped allocation.

Figure 7 – Sun QFS File System Using Mismatched Striped Groups in a Striped Allocation



The following table shows the characteristics of this sample file system.

Table – Sample File System Characteristics

Characteristics	Notes
File system name	`avfs`
Number of stripe groups	Two. The video file group is `g0`. The audio file group is `g1`.
Stripe width	0
DAU	128 kilobytes
Number of disks for `g0`	Eight
Minimum block size for `g0`	Eight disks x 128-kilobyte DAU = 1024 kilobytesThis is the amount of data written in one block write. Each disk receives 128 kilobytes of data, so the total amount written to all disks at one time is 1024 kilobytes.
Number of disks for `g1`	One
Minimum block size for `g1`	One disk x 128-kilobyte DAU = 128 kilobytes

Add the following line to the /etc/vfstab file so that the environment recognizes the avfs file system:

```
avfs    -    /avfs    samfs    -    no    stripe=0
```

Note that in the /etc/vfstab file, stripe=0 is used to specify a round-robin file system. This is used because a value greater than 0 is not supported for mismatched striped groups.

The following example shows the mcf file for file system avfs.

Example – The **mcf** File for File System **avfs**

```
# Equipment Eq Eq Fam Dev Additional
# Identifier Ord Type Set State Parameters
#
avfs 100 ma avfs
/dev/dsk/c00t1d0s6 101 mm avfs -
#
/dev/dsk/c01t0d0s6 102 g0 avfs -
/dev/dsk/c02t0d0s6 103 g0 avfs -
/dev/dsk/c03t0d0s6 104 g0 avfs -
/dev/dsk/c04t0d0s6 105 g0 avfs -
/dev/dsk/c05t0d0s6 106 g0 avfs -
/dev/dsk/c06t0d0s6 107 g0 avfs -
/dev/dsk/c07t0d0s6 108 g0 avfs -
/dev/dsk/c08t0d0s6 109 g0 avfs -
#
/dev/dsk/c09t1d0s6 110 g1 avfs -
```

When the mcf file for this file system is ready, you can enter the sammkfs(1M) and mount(1M) commands shown in the following example to create and mount the avfs file system.

Example – Commands to Create and Mount File System **avfs**

```
# sammkfs -a 128 avfs
# mount avfs
```

After the file system is mounted, you can use the commands shown in the following example to create two directories for the two types of files.

Example – Commands to Create Directories in File System **avfs**

```
# cd /avfs
# mkdir video
# mkdir audio
```

Once the directories are created, you can use the setfa(1) commands shown in the following example to assign the large striped group to video and the small striped group to audio. Files created in these directories are allocated on their respective striped groups because attributes are inherited.

Example – Commands to Set File Attributes

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

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

Per-Logical Unit Number (LUN) Allocation Control

If necessary, you can disable allocation to a specific Sun QFS data partition by using a `nalloc` command, which prohibits any future allocation to that device. The feature is currently only usable on data partitions, not on metadata partitions.

Allocation to a partition can be restarted by either an `alloc` or on command.

The allocation state of a partition (`allocflag`) is persistent across boots.

The `nalloc` and `alloc` commands are available in the `samu` interface, and the samu on command also sets allocation to on. The samu screens display the `nalloc` state for partitions that have been disabled. The `samtrace` and `samfsinfo` output also include the allocation state.

For more information about the `samu` interface, see Using the samu Operator Utility.

Related Topics

- File System Overview
- File System Design Basics

Next Steps

- Configuring the File System
- Configuring a Shared File System

File System Design Basics

Contents

File System Design Basics

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

Sun QFS file systems use a linear search method for directory lookups, searching from the beginning of the directory to the end. As the number of files in a directory increases, the search time through the directory also increases. Search times can become excessive when you have directories with thousands of files. These long search times are also evident when you restore a file system. To increase performance and speed up file system dumps and restores, keep the number of files in a directory under 10,000.

The directory name lookup cache (DNLC) feature improves file system performance. This cache stores the directory lookup information for files whose paths are short (30 characters or less), removing the need for directory lookups to be performed on the fly.

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

Inode Files and File Characteristics

The types of files to be stored in a file system affect file system design. An inode is a 512-byte block of information that describes the characteristics of a file or directory. This information is allocated dynamically within the file system.

Inodes are stored in the `.inodes` file located under the file system mount point.

Like a standard Solaris OS inode, a Sun QFS file system inode contains the file's POSIX standard inode times: file access, file modification, and inode changed times. A Sun QFS file system inode includes other times as well, as shown in the following table.

Table – Content of **`.inode`** Files

Time	Incident

access	Time the file was last accessed. POSIX standard.
modification	Time the file was last modified. POSIX standard.
changed	Time the inode information was last changed. POSIX standard.
attributes	Time the attributes specific to the file system were last changed. Sun Microsystems extension.
creation	Time the file was created. Sun Microsystems extension.
residence	Time the file changed from offline to online or vice versa. Sun Microsystems extension.

> Note -
> If the WORM-FS (write once read many) package is installed, the inode also includes a `retention-end` date. See Configuring WORM-FS File Systems for more information.

For more information on viewing inode file information, see Viewing Files and File Attributes.

Specifying Disk Allocation Units

Disk space is allocated in basic units of online disk storage called disk allocation units (DAUs). Whereas sectors, tracks, and cylinders describe the physical disk geometry, the DAU describes the file system geometry. Choosing the appropriate DAU size and stripe size can improve performance and optimize magnetic disk usage. The DAU setting is the minimum amount of contiguous space that is used when a file is allocated.

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

DAU Settings and File System Geometry

Sun QFS file systems use an adjustable DAU. You can configure the DAU to tune the file system to the physical disk storage device. This feature minimizes the system overhead caused by read-modify-write operations and is therefore particularly useful for applications that manipulate very large files. For information about how to control the read-modify-write operation, see Increasing File Transfer Performance for Large Files.

Each file system can have its own unique DAU setting, even if it is one of several mounted file systems active on a server. The possible DAU settings differ depending on the type of file system you are using. The DAU setting is determined through the `sammkfs(1M)` command when the file system is created. It cannot be changed dynamically.

DAU settings work in conjunction with the device and file system definitions specified in the master configuration (`mcf`) file. For details about the `mcf` file, see About the Master Configuration File and the `mcf(4)` man page.

ms and ***ma*** File Systems

Two file allocation schemes are available to you:

- An `ms` file system type - File system data and file system metadata are on the same device
- An `ma` file system type - File system data and file system metadata are on different devices

For a simple Sun QFS file system, such as one on a single partition, the file system is defined in your `mcf` file by an Equipment Type value of `ms`. In the `ms` file system, the only device type allowed is type `md`, and both metadata and file data are written to the `md` devices. By default, the DAU on an `md` device is 64 kilobytes.

A more complex Sun QFS file system installed on multiple partitions is defined as Equipment Type `ma` in your `mcf` file. In an `ma` file system, metadata is written to `mm` devices, and data can be written to `md`, `mr`, or gXXX devices.

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

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

For more information on these device types, see About the Master Configuration File.

Dual and Single Allocation Schemes

The md and mm devices use a dual allocation scheme, as follows:

- On md data devices, the small allocation is 4 kilobytes, and the large allocation is a DAU. The default DAU is 64 kilobytes. You can override this default when the file system is initialized by using the -a allocation-unit option to the sammkfs(1M) command. The DAU size can be 16, 32, or 64 kilobytes.
- When a file is created on an md device, the system allocates the first eight addresses of the file in the small allocation. If more space is needed, the file system uses one or more large allocations (DAUs) to expand the file. As a result, I/O performance improves for large files while minimizing the disk fragmentation that can result from many small files.

> Note -
> When using an ms file system, the stripe width should be set to greater than zero to stripe metadata information across the disk. However, you should read and understand Stripe Widths on Data Disks before setting the stripe width and DAU size.

- On mm metadata devices, the small allocation is 4 kilobytes, and the large allocation is 16 kilobytes. The dual allocation scheme enables the file system to write metadata to disk more efficiently and helps minimize disk fragmentation.

Depending on the type of file data stored in the file system, a larger DAU size can improve file system performance significantly. For information about tuning file system performance, see Advanced File System Topics.

Only ma Sun QFS file systems can include devices that use a single allocation scheme. These file systems consist of separate metadata devices and data devices, as follows:

- The metadata devices can be defined only as Equipment Type mm.
- The data devices can be defined as Equipment Type md, mr, or gXXX. The md devices are limited to DAU sizes of 16 kilobytes, 32 kilobytes, or 64 kilobytes.
  The mr and gXXX devices follow a single allocation scheme. You can mix mr and gXXX devices in a file system, but you cannot mix md devices with either mr or gXXX devices in a file system. The mr and gXXX devices can be set to a minimum DAU allocation of 8 kilobytes for devices in an ma file system. This setting is optimal for workloads with the majority of file sizes at or below 8 kilobytes.

The DAU size for file systems that use mr and gXXX data devices is configurable. The possible DAU sizes that can be used on data devices depend on the Equipment Type value assigned to each data device in the mcf file. The following table shows these DAU sizes.

Table – Equipment Type Values and DAU Sizes

Equipment Type	DAU Sizes
mr or g XXX	You can specify different DAU sizes by adjusting the default size in 8-kilobyte increments. The DAU size can be from 8 kilobytes to 65,528 kilobytes (64 megabytes). The default DAU size is 64 kilobytes for mr or 256 kilobytes for gXXX.
md	This type of device uses a dual allocation scheme. The DAU can be configured to be 16, 32, or 64 kilobytes in length. The default DAU size is 64 kilobytes.An md device in an ma file system is used to store data only, not metadata. An md device in an ms file system is used to store both file data and metadata.

> Note -
> If you created your file system using version 3.5 of the software, or built it using the sammkfs compatibility mode flag in version 4 of the software, you may be using a version 1 superblock. In the version 1 superblock, mm devices do not use the dual allocation scheme, and the allocation for mm devices is 16 kilobytes. Only a version 2 superblock enables you to define md devices in a Sun QFS file system. To find out whether you are using a version 1 superblock, use the samfsinfo(1M) command.

Data Alignment

Data alignment refers to matching the allocation unit of the RAID controller with the allocation unit of the file system. The optimal file system alignment formula is as follows:

allocation-unit = RAID-stripe-width x number-of-data-disks

For example, suppose a RAID-5 unit has nine disks, with one of the nine being the parity disk, making the number of data disks eight. If the RAID stripe width is 64 kilobytes, then the optimal allocation unit is 64 multiplied by 8, which is 512 kilobytes.

Data files are allocated as striped or round-robin through each striped group (gXXX) or data disk (mr or md) within the same file system.

A mismatched alignment hurts performance because it can cause a read-modify-write operation.

Stripe Widths on Data Disks

Stripe width defaults differ between ms and ma file systems. The stripe width is specified by the -o stripe=n option in the mount(1M) command. If the stripe width is set to 0, round-robin allocation is used.

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

Stripe Widths on **ms** File Systems

On ms file systems, the stripe width is set at mount time. The following table shows default stripe widths.

Table – **ms** File System Default Stripe Widths

DAU	Default Stripe Width	Amount of Data Written to Disk
16 kilobytes	8 DAUs	128 kilobytes
32 kilobytes	4 DAUs	128 kilobytes
64 kilobytes (default)	2 DAUs	128 kilobytes

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

Note -

- To strip metadata information across the disk in an ms file system, set the stripe width to greater than zero.
- If you multiply the number in the first column of the above table by the number in the second column, the resulting number is 128 kilobytes. Sun QFS file systems operate most efficiently if the amount of data being written to disk is at least 128 kilobytes.

Stripe Widths on **ma** File Systems Not Using Striped Groups

On ma file systems, the stripe width that is set at mount time depends on whether or not striped groups are configured. A striped group is a collection of devices that are striped as a group. For more information about striped groups, see File Allocation Methods. This section describes stripe widths for Sun QFS file systems that are configured without stripe groups.

If striped groups are not configured, the DAU and stripe width relationships on ma file systems are similar to those for ms file systems. The difference is that DAUs larger than 64 kilobytes are possible and that the DAU is configurable in 8-kilobyte blocks. The maximum DAU size is 65,528 kilobytes.

By default, if no stripe width is specified, the amount of data written to disk is at or near 128 kilobytes. Sun QFS file systems are most efficient if write operations write at least one whole stripe per I/O request. The following table shows the default stripe widths.

Table – Default Stripe Widths for **ma** File Systems Not Using Striped Groups

DAU	Default Stripe Width	Amount of Data Written to Disk
16 kilobytes	8 DAUs	128 kilobytes
24 kilobytes	5 DAUs	120 kilobytes
32 kilobytes	4 DAUs	128 kilobytes
40 kilobytes	3 DAUs	120 kilobytes
48 kilobytes	2 DAUs	96 kilobytes
56 kilobytes	2 DAUs	112 kilobytes

64 kilobytes (default)	2 DAUs	128 kilobytes
72 kilobytes	1 DAU	72 kilobytes
128 kilobytes	1 DAU	128 kilobytes
> 128 kilobytes	1 DAU	DAU size

Stripe Widths on **ma** File Systems Using Striped Groups

If striped groups are configured for your file system, the minimum amount of space allocated is the DAU multiplied by the number of devices in the striped group. The amount of the allocation can be very large with striped groups.

When striped groups are used, data is written to several disk devices at once, as if they were one device. Allocations on striped groups are equal to the DAU size multiplied by the number of elements in the striped group.

The `-o stripe=n` mount option determines the number of allocations that occur on each stripe group before the allocation moves to a different striped group. If a file system is mounted with `-o stripe=0`, the allocation is always to one striped group.

By default, the setting is `-o stripe=0`, which specifies the round-robin allocation method. The setting can be as low as `-o stripe=0` (which disables striping) or as high as `-o stripe=255`. The system sets `-o stripe=0` if mismatched striped groups are present, in which case a file can reside on only one striped group.

For more information on allocation methods, see File Allocation Methods.

Stripe Widths on Metadata Disks

You can use the `-o mm_stripe=n` option to the `mount_samfs`(1M) command to stripe metadata information on the metadata disk. The default stripe width is `-o mm_stripe=1`, which specifies that the file system write one 16-kilobyte DAU to a metadata disk before switching to the next metadata disk. The small 4-kilobyte DAU is used for metadata disks.

By default, if you have multiple metadata devices, metadata is allocated as specified in the `- o mm_stripe=n` option to the `mount(1M)` command. The setting can be as low as `-o mm_stripe=0`, which disables striping, or as high as `-o mm_stripe=255`.

Related Topics

- File System Overview
- File Allocation Methods

Next Steps

- Configuring the File System
- Configuring a Shared File System

File System Overview

Contents

File System Overview

The Sun QFS file system is a configurable file system that presents a standard UNIX file system interface to users.

General File System Configurations

The Sun QFS file system can be used in several different configurations:

- Configured as a stand-alone file system on a single host
- Configured as a shared file system, in which multiple hosts can write to and read from the file system
- Configured as a multireader file system, in which only one host can write to the file system, but multiple host can read from the file system
- Configured with the Sun Storage Archive Manager (SAM-QFS) product to support archiving features
- Configured with the Sun Cluster product for failover and high-availability

Unless otherwise noted, file system information throughout this space applies to archiving and non-archiving configurations.

The Sun QFS file system does not require changes to user programs or to the UNIX kernel.

File System Features

Sun QFS file systems provide the following key features:

- Supports flexible volume management
- Supports paged and direct I/O
- Supports very large files
- Provides fast file system recovery
- Enables flexible metadata storage
- Uses virtual file system `vfs/vnode` interface
- Supports shared file systems

Volume Management

Sun QFS file systems support both striped and round-robin disk access. The master configuration file (`mcf`) and the mount parameters specify the volume management features and enable the file system to recognize the relationships between the devices it controls. This is in contrast to most UNIX file systems, which can address only one device or one portion of a device. Sun QFS file systems do not require additional volume management applications. However, if you want to use mirroring for devices in a Sun QFS environment, you must obtain an additional package, such as a logical volume manager.

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

Support for Paged and Direct I/O

The Sun QFS file system supports two different types of I/O:

- Paged I/O – Also known as cached or buffered I/O. When paged I/O is used, user data is cached in virtual memory pages and the kernel writes the data to disk. The standard Solaris OS interfaces manage paged I/O. This is the default type of I/O for Sun QFS .
- Direct I/O – When direct I/O is used, user data is written directly from user memory to disk. You can specify direct I/O by using the Solaris OS `directio`(3C) function call or the `setfa`(1) command with its `-D` option. By using direct I/O, you can realize substantial performance improvements for large block, sequential, aligned I/O.

High Capacity

The Sun QFS software supports files of up to 2^{63} bytes in length. Such very large files can be striped across many disks or RAID devices, even within a single file system. This is true because Sun QFS file systems use true 64-bit addressing, in contrast to standard UNIX file systems, which are not true 64-bit file systems.

The number of file systems that you can configure is virtually unlimited. The volume manager enables each file system to include up to 252 device partitions, typically disk. Each partition can include up to 16 terabytes of data. This configuration offers virtually unlimited storage capacity.

There is no predefined limit on the number of files in a Sun QFS file system. Because the inode space (which holds information about the files) is dynamically allocated, the maximum number of files is limited only by the amount of disk storage available. The inodes are cataloged in the `.inodes` file under the mount point. The `.inodes` file requires a minimum of 512 bytes of storage per file.

In a Sun QFS file system, the inodes are located on the metadata devices and can be separated from the file data devices. In practice, the size of your metadata (mm) devices limits the number of files in a Sun QFS file system, but you can increase the maximum number of files by adding more metadata devices. The hard limit on the number of files is $2^{32}-1$ files, and the recommended limit is 10^{8} files.

Fast File System Recovery

A key function of a file system is its ability to recover quickly after an unscheduled outage. Standard UNIX file systems require a lengthy file system check (`fsck`(1M)) to repair inconsistencies after a system failure.

A Sun QFS file system often does not require a file system check after a disruption that prevents the file system from being written to disk (using `sync`(1M)). In addition, Sun QFS file systems recover from system failures without using journaling. They accomplish this dynamically by using identification records, serial writes, and error checking for all critical I/O operations. After a system failure, even multiterabyte-sized Sun QFS file systems can be remounted immediately.

Metadata Storage

File systems use metadata to reference file and directory information. Typically, metadata resides on the same device as the file data. However, the Sun QFS file system has the option of separating the file system metadata from the file data by storing them on separate devices. The Sun QFS file system enables you to define one or more separate metadata devices in order to reduce device head movement and rotational latency, improve RAID cache utilization, or mirror metadata without mirroring file data.

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

vnode Interface

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

By using the `vfs/vnode` interface, the file system works with the standard Solaris OS kernel and requires no modifications to the kernel for file management support. Thus, the file system is protected from operating system changes and typically does not require extensive regression testing when the operating system is updated.

The kernel intercepts all requests for files, including those that reside in Sun QFS file systems. If a file is identified as a Sun QFS file, the kernel passes the request to the appropriate file system for handling. Sun QFS file systems are identified as type `samfs` in the `/etc/vfstab` file and through the `mount`(1M) command.

Shared File System Support

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

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

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

Sun QFS shared file systems do not support the following:

- These file types:
  - b– Block special files
  - c– Character special files
  - p– FIFO (named pipe) special files
- Segmented files. You cannot implement a SAM-QFS shared file system in a segmented-file environment.
- Mandatory locks. An EACCES error is returned if the mandatory lock is set. Advisory locks are supported, however. For more information about advisory locks, see the `fcntl`(2) system call.

For more information about shared file systems, see Configuring a Shared File System.

Additional File System Features

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

- Application programming interface (API) routines– API routines enable a program to perform various specialized functions, such as preallocating contiguous disk space or accessing a specific striped group. For more information about these routines, see the `intro_libsam(3)` man page.
- Adjustable disk allocation units (DAUs)– The DAU is the basic unit of online storage. The Sun QFS file system software includes an adjustable DAU, which is useful for tuning file systems with the physical disk storage device and for eliminating the system overhead caused by read-modify-write operations. For information about adjusting the DAU size, see Specifying Disk Allocation Units.
- Support for multiple striped groups– To support multiple RAID devices in a single file system, Sun QFS software supports the definition of striped groups. You can optimize disk block allocation for a striped group, thereby reducing the overhead for updating the on-disk allocation map. Users can assign a file to a striped group either through an API routine or by using the `setfa(1)` command.

Related Topics

- File System Design Basics
- File Allocation Methods

Next Steps

- Configuring the File System
- Configuring a Shared File System

Mount Options in a Shared File System

Contents

Mount Options in a Shared File System

The Sun QFS shared file system can be mounted with several mount options. This section describes some of these options within the context of their roles.

You can specify most mount options by using the `mount`(1M) command, by entering them in the `/etc/vfstab` file, or by entering them in the `samfs.cmd` file. For example, the following `/etc/vfstab` file includes `mount`(1M) options for a shared file system:

```
sharefs1 - /sfs samfs - no shared,mh_write
```

You can change some mount options dynamically by using the `samu`(1M) operator utility. For more information about these options, see Using the samu Operator Utility.

The following sections summarize the mount options available to you in a shared file system. For more information about any of these mount options, see the `mount_samfs`(1M) man page or see the cross-references mentioned in their descriptions.

Mounting in the Background: the **bg** Option

The bg mount option specifies that if the first mount operation fails, subsequent attempts at mounting should occur in the background. By default, bg is not in effect, and mount attempts continue in the foreground.

Reattempting a File System Mount: the **retry** Option

The `retry` mount option specifies the number of times that the system should attempt to mount a file system. The default is 10000.

Declaring a Sun QFS Shared File System: the **shared** Option

The `shared` mount option declares a file system to be a Sun QFS shared file system. This option must be specified in the `/etc/vfstab` file in order for the file system to be mounted as a Sun QFS shared file system. The presence of this option in a `samfs.cmd` file or on the `mount`(1M) command does not cause an error condition, but it does not mount the file system as a shared file system.

Tuning Allocation Sizes: the **minallocsz=**n and **maxallocsz=**n Options

The `-o minallocsz=`n and `-o maxallocsz=`n options to the `mount`(1M) command specify an amount of space, in kilobytes. This is the minimum block allocation size. If a file is growing, the metadata server allocates blocks when an append lease is granted. You can use the `-o minallocsz=`n option to specify the initial size of this allocation. The metadata server can increase the size of the block allocation depending on the application's access patterns up to, but not exceeding, the `-o maxallocsz=`n option's setting.

You can specify these `mount`(1M) options on the `mount`(1M) command line, in the `/etc/vfstab` file, or in the `samfs.cmd` file.

Using Leases in a Sun QFS Shared File System: the **rdlease=**n, **wrlease=**n, and **aplease=**n Options

A lease grants a shared host permission to perform an operation on a file for as long as the lease is valid. The metadata server issues leases to each shared host, including itself. The leases are renewed as necessary to permit continued file operations. The possible file operations are as follows:

- A read lease enables existing file data to be read.
- A write lease enables existing file data to be overwritten.
- An append lease enables a file's size to be extended and enables newly allocated blocks to be written.

A shared host can continue to update leases for as long as necessary. The lease is transparent to the end user. The following table shows the mount options that enable you to specify the duration of each lease type.

Table – Lease-Related **mount**(1M) Options

Option	Action
`-o rdlease=n`	This option specifies the maximum amount of time, in seconds, for the read lease.
`-o wrlease=n`	This option specifies the maximum amount of time, in seconds, for the write lease.
`-o aplease=n`	This option specifies the maximum amount of time, in seconds, for the append lease.

All three leases enable you to specify an n such that $15 \leq n \leq 600$. The default time for each lease is 30 seconds. A file cannot be truncated if a lease is in effect. For more information about setting these leases, see the `mount_samfs`(1M) man page.

If you change the metadata server because the current metadata server is down, you must add the lease time to the changeover time because all leases must expire before an alternate metadata server can assume control.

Setting a short lease time causes more traffic between the client hosts and the metadata server because the lease must be renewed after it has expired.

Enabling Multiple Host Reads and Writes: the **mh_write** Option

By default, in a Sun QFS shared file system, multiple hosts can read the same file at the same time, and if no host is writing to that file, I/O can be paged on all hosts. Only one host can append or write to a file at any one time.

The mh_write option controls write access to the same file from multiple hosts. If mh_write is specified as a mount option on the metadata server host, the Sun QFS shared file system enables simultaneous reads and writes to the same file from multiple hosts. If mh_write is not specified on the metadata server host, only one host can write to a file at any one time.

By default, mh_write is disabled, and only one host has write access to a file at any one time. The length of that time period is determined by the duration of the wrlease mount option. If the Sun QFS shared file system is mounted on the metadata server with the mh_write option enabled, simultaneous reads and writes to the same file can occur from multiple hosts.

The following table describes how file access from multiple hosts is affected depending on whether the mh_write option is enabled on the metadata server.

Table – File Access Based on the **mh_write** Option

mh_write Not Enabled on the Metadata Server	**mh_write** Enabled on the Metadata Server
Multiple reader hosts allowed.Can use paged I/O.	Multiple reader hosts allowed.Can use paged I/O.
Only one writer host is allowed.Can use paged I/O.All other hosts wait.	Multiple reader and/or writer hosts allowed.If any writer hosts exist, all I/O is direct.
Only one append host.All other hosts wait.	Only one append host is allowed.All other hosts can read and/or write.If any writer hosts exist, all I/O is direct.

The mh_write option does not change locking behavior. File locks behave the same whether mh_write is in effect or not. The mh_write option's effect is as follows:

- When mh_write is in effect, all hosts can read from and write to the same file simultaneously.
- When mh_write is not in effect, only one host can write to a given file during a given time interval, and no hosts can read from the file during that time interval.

Sun QFS shared file system maintains consistency between hosts. The first time that a host executes a read or write system call, it gets a lease, which enables it to read or write the file for some period of time. The existence of that lease prevents other hosts without mh_write from accessing the file. In particular, the lease can last longer than the duration of the system call that caused its acquisition.

When mh_write is not in effect, the Sun QFS shared file system should provide near-POSIX behavior for data reads and writes. For metadata, however, access time changes might not be seen immediately on other hosts. Changes to a file are pushed to disk at the end of a write lease, and when a read lease is acquired, the system invalidates any stale cache pages so that the newly written data can be seen.

When mh_write is in effect, behavior might be less consistent. When there are simultaneous readers and writers, the Sun QFS shared file system switches all hosts accessing the file into direct I/O mode. This means that page-aligned I/O should be visible immediately to other hosts. However, non-page-aligned I/O can result in stale data being visible, or even written to the file, because the normal lease mechanism that prevents this has been disabled.

You should specify the mh_write option only when multiple hosts need to write to the same file simultaneously and when applications perform page-aligned I/O. In other cases, there is some risk of data inconsistency because even using flock() (which works with mh_write) to coordinate between hosts does not guarantee consistency.

For more information about mh_write, see the mount_samfs(1M) man page.

Setting the Minimum Number of Concurrent Threads: the **min_pool=**n Option

The min_pool=n mount option sets the minimum number of concurrent threads for the Sun QFS shared file system. By default, min_pool=64 on Solaris systems. This means that using default settings, there will always be at least 64 active threads in the thread pool on Solaris and 8 on Linux. You can adjust the min_pool=n mount option to any value between 8 and 2048, depending on the Sun QFS shared file system's activity.

The min_pool mount option must be set in the samfs.cmd file. It will be ignored if set in the /etc/vfstab file or on the command line.

> Note -
> The min_pool mount option replaces the previous nstreams mount option. In version 5.0 of the software, the nstreams option is completely removed.

Retaining Cached Attributes: the **meta_timeo=**n Option

The meta_timeo=n mount option determines how long the system waits between checks on the metadata information. By default, the system refreshes metadata information every three seconds. This means, for example, that an ls(1) command entered in a Sun QFS shared file system with several newly created files might not return information about all the files until three seconds have passed. For n, specify a value such that 0≤ n≤ 60.

Specifying Striped Allocation: the **stripe** Option

By default, data files in the Sun QFS shared file system are allocated using the round-robin file allocation method. To specify that file data be striped across disks, you can specify the stripe mount option on the metadata host and all potential metadata hosts. Note that by default, unshared file systems allocate file data using the striped method.

In a round-robin allocation, files are created in a round-robin fashion on each slice or striped group. This causes the maximum performance for one file to be the speed of a slice or striped group. For more information about file allocation methods, see File System Design Basics.

Specifying the Frequency With Which Metadata Is Written: the **sync_meta=**n Option

You can set the sync_meta=n option to sync_meta=1 or sync_meta=0.

By default, sync_meta=1 and a Sun QFS shared file system writes file metadata to disk every time the metadata changes. This slows data performance, but it ensures data consistency. This is the setting that must be in effect if you want to change the metadata server.

If you set sync_meta=0, the Sun QFS shared file system writes the metadata to a buffer before writing it to disk. This delayed write delivers higher performance, but it decreases data consistency after an unscheduled machine interruption.

Enabling WORM Functionality: the **worm_capable** and **def_retention** Options

If you are using the optional WORM package, the worm_capable mount option enables the file system to support WORM files. The def_retention mount option sets the default retention time using the format {{def_retention=}}MyNdOhPm.

In this format, M, N, O, and P are non-negative integers and y, d, h, and m stand for years, days, hours, and minutes, respectively. Any combination of these units can be used. For example, 1y5d4h3m indicates 1 year, 5 days, 4 hours, and 3 minutes; 30d8h indicates 30 days and 8 hours; and 300m indicates 300 minutes. This format is backward compatible with the formula in previous software versions, in which the retention period was specified in minutes.

See Configuring WORM-FS File Systems for more information about the WORM functionality.

Performing Operations

Contents

Performing Operations

This section contains information about file system operations that you might need to perform on a daily basis.

Associating Solaris Projects With Files and Directories

Beginning with Sun QFS 5.0, you can associate standard Solaris projects with your files and directories.

Identifying the Project for a File System

Use one of the following command options to see the project that is associated with a file or directory:

Command Option	Description	More Information
`sls -D`	The `sls -D` command displays a file's project name and ID. This project functionality is available on all Linux and Solaris hosts.	See the `sls`(1) man page.
`sfind -project` project-name | project-id	The `sfind -project` command can be used to find all files with the specified project name or project ID within a given file hierarchy. This project functionality is available on all Linux and Solaris hosts.	See the `sfind`(1) man page.

Example – Viewing Project Information for a File

```
# sls -D mickey.gif
  mickey.gif:
     mode: -rw-r--r--  links:   1  owner: root      group: other
     length:    319279  admin id:      7  inode: 1407.5
     project: system(0)
     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
```

Example – Finding Files Associated With a Project ID

```
# sfind -project myproject
```

Changing the Project Association (**`schproj`**(1) command)

> Note -
> The `schproj`(1) command does not apply to Linux clients.

After you have create a file or directory, use the `schproj` command to change the associated project. The `schproj`(1) command options are the same as those for the `chgrp`(1) command, with identical behavior for files, directories, hard links, and symbolic links. Available project names and IDs are defined in the Solaris file `/etc/project`. The project names and IDs that are valid for a user task to set on a file or directory can be viewed using the `projects`(1) command. The root user may set on a file or directory any available project that is defined in `/etc/project`. The task current active project that will be automatically assigned to a new file or directory upon its creation can be viewed using `id -p`.

Example – Changing Project Association

```
# schproj myproject myfilename
```

Viewing Files and File Attributes

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

File Attributes and File States

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

To set attributes, use the following commands:

- `archive`(1)
- `ssum`(1)
- `release`(1)
- `segment`(1)
- `setfa`(1)
- `stage`(1)

To set attributes from within applications, use the following application programming interface (API) routines:

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

The following table shows the user-specified attributes that are listed in the inode.

Table – User-Specified File Attributes

File Attribute	Description
`archive -C`	Marks the file for concurrent archiving. This means that the file can be archived even if it is open for a write operation. You can use the `archive`(1) command to set this attribute.
`archive -n`	Marks the file to never be archived. The superuser can use the `archive`(1) command to set this attribute.
`release -a`	Marks the file to be released as soon as one archive copy is made. You can set this attribute from within the `archiver.cmd` file or by using the `release`(1) command.
`release -n`	Marks the file to never be released. You can set this attribute from within the archiver.cmd file, or the superuser can use the release(1) command to set it.
`release -p`	Marks the file for partial release. You can set this attribute from within the `archiver.cmd` file or by using the `release`(1) command.
`stage -a`	Marks the file for associative staging. You can set this attribute from within the `archiver.cmd` file or by using the `stage` (1) command.
`stage -n`	Marks the file to never be staged. This signifies direct access to removable media cartridges. You can set this attribute from within the `archiver.cmd` file, or the superuser can use the `stage`(1) command to set it.Not supported on Sun QFS shared file system clients.
`setfa -D`	Marks the file for direct I/O.
`setfa -gn`	Marks the file for allocation on striped group n.
`setfa -sm`	Marks the file for allocation with a stripe width of m.
`segment` nm `stage_ahead` x	Marks the file for segmentation. The nm notation indicates that the segment is n megabytes in size. The `stage_ahead` x attribute indicates the number of segments (x) to be staged ahead. You can use the `segment`(1) command to set this attribute.

Note -
There are several additional user-specified file attributes that are specific to archiving functionality. See the Sun Storage Archive Manager Configuration and Administration Guide for more information.

You can set the attributes shown in the above table on both files and directories. After directory attributes are set, files that are created in the directory inherit all the directory attributes. Files created before an attribute is applied to the parent directory do not inherit directory attributes.

If you have the WORM-FS package (SUNWsamfswm) installed, you can also apply WORM (write once read many) attributes to a file, and set the file's retention period. See Configuring WORM-FS File Systems for details.

Displaying File Information

The `sls(1)` command extends the standard UNIX `ls(1)` command and provides more information about a file. The following example shows detailed `sls(1)` command output that displays the inode information for file `hgc2`.

Example – **`sls`** Output in a SAM-QFS Environment

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

The following table describes the meaning of each row of `sls(1)` output shown in the example.

Note -
Lines that pertain to archiving only appear in `sls(1)` output only in a Sun SAM environment.

Table – **`sls`**(1) Output Explanation

Line Number	Tag	Content
1	`mode:`	The file's mode and permissions, the number of hard links to the file, the owner of the file, and the group to which the owner belongs.
2	`length:`	The file's length in bytes, the file's admin ID number, and the file's inode number.By default, the admin ID number is 0. If this number is greater than 0, it indicates the file's accounting category for counting files and blocks. You can set this number to a value greater than 0 even when file system quotas are not enabled on this file system. For information about file system quotas, see Administering File System Quotas.The inode number is a two-part number that contains the inode number itself, followed by a period (.), followed by the inode generation number.
3	`archdone;`	The file attributes specific to the file. For more information about this line, see the `sls(1)` man page.
4	`segments`	The segment index information. This line does not appear unless the file is a segment index. The general format for this line is as follows:`segments` n`,` `offline` o`,` `archdone` a`,` `damaged` d`;` • `segments` n shows the total number of data segments for this file. In this example, there are 3. • `offline` o shows the number of data segments offline. In this example, there are no offline segments. • `archdone` a shows the number of segments for which the archiving requirements have been met. In this example, there are 3. • `damaged` d shows the number of damaged segments. In this example, there are no damaged segments.
5, 6	`copy 1:`, `copy 2:`	Archive copy lines. The `sls(1)` command displays one archive copy line for each active or expired archive copy.The four positions in this line indicate the following:

		1 – Either an expired or an active entry. • An S indicates that the archive copy is expired. That is, the file was modified and this archive copy is a previous version of the file. • A U indicates that the copy has been unarchived. Unarchiving is the process by which archive entries for files or directories are deleted. • A dash (`-`) indicates that the archive copy is active and valid.
		2 – Whether the archive copy is to be rearchived. • An r indicates that the archive copy is scheduled to be rearchived by the archiver. • A dash (`-`) indicates that the archive copy is not to be rearchived by the archiver.
		3 – Unused.
		4 – Whether the copy is damaged or undamaged. • A D indicates that the archive copy is damaged. A damaged archive copy is not a candidate for staging. • A dash (`-`) indicates that the archive copy is not damaged. It is a candidate for staging.
		The format of the rest of the archive copy line is as follows: • The date and time the archive copy was written to the archive media. • Two hexadecimal numbers separated by a decimal point (.). The first hexadecimal number (`2239a`) indicates the position of the beginning of the archive file on the cartridge. The second hexadecimal number (`48`) is the file byte offset (divided by 512) of this copy in the archive file. • The media type and the volume serial name (VSN) where the archive copy resides.
7	`access:`	The time the file was last accessed and modified.
8	`changed:`	The time the file content and the file's attributes were last changed.
9	`creation:`	The time the file was created and became resident in the file system.

About the Retention Line

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

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

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

Propagating Configuration File Changes to the System

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

- `mcf`(4)
- `defaults.conf`
- `archiver.cmd` (archiving file systems only)
- `stager.cmd` (archiving file systems only)
- shared hosts file (shared file systems only)

You must perform these procedures under the following circumstances:

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

The following sections describe these procedures.

How to Change **mcf** or **defaults.conf** File System Information in a SAM-QFS Environment

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

```
# sam-fsd
```

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

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

```
samcmd aridle fs._fsname_
```

   For fsname, specify the name of the file system.

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

```
samcmd idle _eq_
```

   For eq, specify the Equipment Number number of the drive.

5. Issue the `umount(1M)` command to unmount each file system affected by the changes.
   For instructions on unmounting the file system, see Unmounting a File System.
6. Use the `samd(1M)` `config` command to propagate the changes:

```
# samd config
```

7. Use the `mount(1M)` command to remount the file systems you unmounted.
   For more information about these files, see the `defaults.conf(4)` or `mcf(4)` man pages.

How to Change **mcf** or **defaults.conf** Removable Media Drive Information

1. Edit the file and change the removable media drive information.
2. If you are changing the `mcf` file, use the `sam-fsd(1M)` command to check the `mcf` file for errors:

```
# sam-fsd
```

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

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

```
samcmd aridle fs.<fsname>
```

   For fsname, specify the name of the file system.

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

```
# samcmd aridle
# samcmd stidle
```

Wait until the tape drives are idle. Then unload the tape drives:

For eq, specify the Equipment Number of the drive.

1. Use the samd(1M) unload command to unload all removable media:

```
# samcmd unload <eq>
```

Repeat the above commands for each tape drive.

Once the tape drives are empty issue the samd stop command:

```
# samd stop
```

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

```
# samd config
```

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

```
# samd start
```

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

Changing the Shared Hosts File

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

How to Add New Entries or Change Existing Entries

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

1. If you do not know the host that is acting as the metadata server, issue the samsharefs(1M) family-set-name command to display the name of the metadata server.
   Issue this command from any host that has the file system configured.
   For example:

```
# samsharefs sharefs1
```

2. On the metadata server, copy the shared hosts file.
   For example:

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

3. Use vi(1) or another editor to edit the shared hosts file.
   For mounted file systems, you can add new host entries to the end of the file, and you can make changes to columns 2 through 5 for

existing entries.
4. Save and close the shared hosts file.
5. Apply the new shared hosts file to the file system.
   For example:

```
# samsharefs -u sharefs1
```

How to Change Host Names, Reorder Entries, or Insert Entries

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

1. If you do not know the host that is acting as the metadata server, issue the `samsharefs`(1M) `-R` family-set-name command to display the name of the metadata server.
   Issue this command from any host that has the file system configured.
   For example:

```
# samsharefs -R sharefs1
```

2. Unmount the file system on each participating client, and then on the metadata server.
3. On the metadata server, copy the shared hosts file.
   For example:

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

4. Use `vi`(1) or another editor to edit the shared hosts file.
5. Save and close the shared hosts file.
6. Apply the new shared hosts file to the file system.
   For example:

```
# samsharefs -uR sharefs1
```

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

Setting Up Mount Parameters

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

Mount parameters are used to manipulate file system characteristics. There are several ways to specify mount parameters. Methods at the top of the hierarchy override methods lower in the hierarchy. You can specify mount options in the following ways, listed in order of precedence with the `mount` command overriding all other settings:

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

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

The following subsections describe ways to specify mount options. For information about specific mount options, see Mount Options in a Shared File System and the `mount_samfs`(1M) man page.

The **mount**(1M) Command

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

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

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

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

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

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

The **/etc/vfstab** File

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

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

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

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

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

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

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

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

The **samfs.cmd** File

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

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

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

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

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

The following example shows a sample `samfs.cmd` file that sets the low-water and high-water marks for disk cache utilization for all file systems and specifies individualized parameters for two specific file systems.

Example – Example **`samfs.cmd`** File

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

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

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

Unmounting a File System

Use the Solaris OS `umount`(1M) command to unmount the file systems.

How to Unmount a File System

Before You Begin

- If the file system has been shared through NFS, use the `unshare`(1M) command to unshare it before you unmount it.
- If the file system is a Sun QFS shared file system, you must unmount the file system on each client before you unmount the file system on the metadata server. For more information, see How to Unmount a Shared File System.

Step

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

  ```
  # umount /samqfs
  ```

  If the command fails and you still need to unmount the file system, use the `umount -f` option to force the file system to unmount.

Related Topics

- Recreating a File System
- Resizing File Systems

Tunable Parameters

Contents

Tunable Parameters

This section describes various ways that you can improve file system and archiving performance.

> Note -
> Sun recommends that you experiment with performance tuning outside of a production environment. Tuning these variables incorrectly can have unexpected effects on the overall system.
> If your site has a Sun™ Enterprise Services (SES) support contract, please inform SES if you change performance tuning parameters.

Increasing File Transfer Performance for Large Files

Sun QFS file systems are tuned to work with a mix of file sizes. You can increase the performance of disk file transfers for large files by enabling file system settings.

How to Increase File Transfer Performance

1. Set the maximum device read/write directive.
   The `maxphys` parameter in the Solaris `/etc/system` file controls the maximum number of bytes that a device driver reads or writes at any one time. The default value for the `maxphys` parameter can differ, depending on the level of your Sun Solaris OS, but it is typically around 128 kilobytes.
   Add the following line to `/etc/system` to set `maxphys` to 1 megabytes:

   ```
   set maxphys = 0x100000
   ```

   > Note -
   > The `maxphys` value must be set to a power of two.

2. Set the SCSI disk maximum transfer parameter.
   The `sd` driver enables large transfers for a specific file by looking for the `sd_max_xfer_size` definition in the `/kernel/drv/sd.conf` file. If this definition does not exist, the driver uses the value defined in the `sd` device driver definition, `sd_max_xfer_size`, which is 1024 x 1024 bytes.
   To enable and encourage large transfers, add the following line at the end of the `/kernel/drv/sd.conf` file:

   ```
   sd_max_xfer_size=0x800000;
   ```

3. Set the fibre disk maximum transfer parameter.
   The `ssd` driver enables large transfers for a specific file by looking for the `ssd_max_xfer_size` definition in the `/kernel/drv/ssd.conf` file. If this definition does not exist, the driver uses the value defined in the `ssd` device driver definition, `ssd_max_xfer_size`, which is 1024 x 1024 bytes.
   Add the following line at the end of the `/kernel/drv/ssd.conf` file:

   ```
   ssd_max_xfer_size=0x800000;
   ```

> **Note -**
> On Solaris 10 x86 platforms, this change is made in the `/kernel/drv/sd.conf` file. For a maximum transfer size of 8 MBytes, the following line is added.`sd_max_xfer_size=0x800000`

4. Reboot the system.
5. Set the `writebehind` parameter.
   This step affects paged I/O only.
   The `writebehind` parameter specifies the number of bytes that are written behind by the file system when paged I/O is being performed on a Sun QFS file system. Matching the `writebehind` value to a multiple of the RAID's read-modify-write value can increase performance.
   This parameter is specified in units of kilobytes and is truncated to an 8-kilobyte multiple. If set, this parameter is ignored when direct I/O is performed. The default `writebehind` value is 512 kilobytes. This value favors large-block, sequential I/O.
   Set the `writebehind` size to a multiple of the RAID 5 stripe size for both hardware and software RAID-5. The RAID-5 stripe size is the number of data disks multiplied by the configured stripe width.
   For example, assume that you configure a RAID-5 device with three data disks plus one parity disk (3+1) with a stripe width of 16 kilobytes. The `writebehind` value should be 48 kilobytes, 96 kilobytes, or some other multiple, to avoid the overhead of the read-modify-write RAID-5 parity generation.
   For Sun QFS file systems, the DAU (`sammkfs`(1M) `-a` command) should also be a multiple of the RAID-5 stripe size. This allocation ensures that the blocks are contiguous.
   You should test the system performance after resetting the `writebehind` size. The following example shows testing timings of disk writes:

   ```
   # timex dd if=/dev/zero of=/sam/myfile bs=256k count=2048
   ```

   You can set the `writebehind` parameter from a mount option, from within the `samfs.cmd` file, from within the `/etc/vfstab` file, or from a command within the `samu`(1M) utility. For information about enabling this from a mount option, see the{{– o writebehind=}} n option on the `mount_samfs`(1M) man page. For information about enabling this from the `samfs.cmd` file, see the `samfs.cmd`(4) man page. For information about enabling this from within `samu`(1M), see the `samu`(1M) man page.

6. Set the `readahead` parameter.
   This step affects paged I/O only.
   The `readahead` parameter specifies the number of bytes that are read ahead by the file system when paged I/O is being performed on a Sun QFS file system. This parameter is specified in units of kilobytes and is truncated to an 8-kilobyte multiple. If set, this parameter is ignored when direct I/O is performed.
   Increasing the size of the `readahead` parameter increases the performance of large file transfers, but only to a point. You should test the performance of the system after resetting the `readahead` size until you see no more improvement in transfer rates. The following is an example method of testing timings on disk reads:

   ```
   # timex dd if=/sam/myfile of=/dev/null bs=256k
   ```

   You should test various `readahead` sizes for your environment. The `readahead` parameter should be set to a size that increases the I/O performance for paged I/O, but is not so large as to hurt performance. It is also important to consider the amount of memory and number of concurrent streams when you set the `readahead` value. Setting the `readahead` value multiplied by the number of streams to a value that is greater than memory can cause page thrashing.
   The default `readahead` value is 1024 kilobytes. This value favors large-block, sequential I/O. For short-block, random I/O applications, set `readahead` to the typical request size. Database applications do their own read-ahead, so for these applications, set `readahead` to 0.
   The `readahead` setting can be enabled from a mount option, from within the `samfs.cmd` file, from within the `/etc/vfstab` file, or from a command within the `samu`(1M) utility. For information about enabling this setting from a mount option, see the{{– o readahead=}}n option on the `mount_samfs`(1M) man page. For information about enabling this setting from the `samfs.cmd` file, see the `samfs.cmd`(4) man page. For information about enabling this setting from within `samu`(1M), see the `samu`(1M) man page.

7. Set the stripe width.
   The{{– o stripe=}}n option with the `mount`(1M) command specifies the stripe width for the file system. The stripe width is based on the disk allocation unit (DAU) size. The n argument specifies that n x DAU bytes are written to one device before writing switches to the next device. The DAU size is set when the file system is initialized by the `sammkfs`(1M) `-a` command.
   If `-o stripe=0` is set, files are allocated to file system devices using the round-robin allocation method. With this method, each file is

completely allocated on one device until that device is full. Round-robin is the preferred setting for a multistream environment. If{{– o stripe=}}n is set to an integer greater than 0, files are allocated to file system devices using the stripe method. To determine the appropriate `-o stripe=`n setting, try varying the setting and taking performance readings. Striping is the preferred setting for turnkey applications with a required bandwidth.
You can also set the stripe width from the `/etc/vfstab` file or from the `samfs.cmd` file.
For more information about the `mount`(1M) command, see the `mount_samfs`(1M) man page. For more information about the `samfs.cmd` file, see the `samfs.cmd`(4) man page.

Enabling Qwrite Capability

By default, the Sun QFS file systems disable simultaneous reads and writes to the same file. This is the mode defined by the UNIX `vnode` interface standard, which gives exclusive access to only one write while other writers and readers must wait. Qwrite enables simultaneous reads and writes to the same file from different threads.

The Qwrite feature can be used in database applications to enable multiple simultaneous transactions to the same file. Database applications typically manage large files and issue simultaneous reads and writes to the same file. Unfortunately, each system call to a file acquires and releases a read/write lock inside the kernel. This lock prevents overlapped (or simultaneous) operations to the same file. If the application itself implements file-locking mechanisms, the kernel-locking mechanism impedes performance by unnecessarily serializing I/O.

Qwrite can be enabled in the `/etc/vfstab` file, in the `samfs.cmd` file, and as a mount option. The `-o qwrite` option with the `mount`(1M) command bypasses the file system locking mechanisms (except for applications accessing the file system through the network file system (NFS)) and lets the application control data access. If `qwrite` is specified, the file system enables simultaneous reads and writes to the same file from different threads. This option improves I/O performance by queuing multiple requests at the drive level.

The following example uses the `mount`(1M) command to enable Qwrite on a database file system:

```
# mount -F samfs -o qwrite /db
```

For more information about this feature, see the `qwrite` directive on the `samfs.cmd`(4) man page or the `-o qwrite` option on the `mount_samfs`(1M) man page.

Setting the Write Throttle

The `-o wr_throttle=`n option limits the number of outstanding write kilobytes for one file to n. By default, Sun QFS file systems set the `wr_throttle` to 16 megabytes.

If a file has n write kilobytes outstanding, the system suspends an application that attempts to write to that file until enough bytes have completed the I/O to allow the application to be resumed.

If your site has thousands of streams, such as thousands of NFS-shared workstations accessing the file system, you can tune the `-o wr_throttle=`n option in order to avoid flushing excessive amounts of memory to disk at once. Generally, the number of streams multiplied by 1024 x the n argument to the `-o wr_throttle=`n option should be less than the total size of the host system's memory minus the memory needs of the Solaris OS, as shown in this formula:
number-of-streams x n x 1024 < total-memory - Solaris-OS-memory-needs

For turnkey applications, you might want to use a size larger than the default 16,384 kilobytes, because this keeps more pages in memory.

Setting the Flush-Behind Rate

Two mount parameters control the flush-behind rate for pages written sequentially and for stage pages. The `flush_behind` and `stage_flush_behind` mount parameters are read from the `samfs.cmd` file, the `/etc/vfstab` file, or the `mount`(1M) command.

The `flush_behind=`n mount parameter sets the maximum flush-behind value. Modified pages that are being written sequentially are written to disk asynchronously to help the Solaris™ Volume Manager (SVM) layer keep pages clean. To enable this feature, set n to be an integer from 16 through 8192. By default, n is set to 0, which disables this feature. The n argument is specified in kilobyte units.

The `stage_flush_behind=`n mount parameter sets the maximum stage flush-behind value. Stage pages that are being staged are written to disk asynchronously to help the SVM layer keep pages clean. To enable this feature, set n to be an integer from 16 through 8192. By default, n is set to 0, which disables this feature. The n argument is specified in kilobyte units.

For more information about these mount parameters, see the `mount_samfs`(1M) man page or the `samfs.cmd`(4) man page.

Tuning the Number of Inodes and the Inode Hash Table

The Sun QFS file system enables you to set the following two tunable parameters in the `/etc/system` file:

- `ninodes`
- `nhino`

To enable non-default settings for these parameters, edit the `/etc/system` file, and then reboot your system.

The following subsections describe these parameters in more detail.

The **ninodes** Parameter

The `ninodes` parameter specifies the maximum number of default inodes. The value for `ninodes` determines the number of in-core inodes that Sun QFS software keeps allocated to itself, even when applications are not using many inodes.

The format for this parameter in the `/etc/system` file is as follows:

```
set samfs:ninodes = _value_
```

The range for value is from 16 through 2000000. The default value for `ninodes` is one of the following:

- A value that is equal to the `ncsize` setting. The `ncsize` parameter is a Solaris tuning parameter that specifies the number of entries in the directory name look-up cache (DNLC). For more information about `ncsize`, see the Solaris Tunable Parameters Reference Manual.
- `2000`. The file systems set `ninodes` to `2000` if the `ncsize` setting is zero or out of range.

The **nhino** Parameter

The `nhino` parameter specifies the size of the in-core inode hash table.

The format for this parameter in the `/etc/system` file is as follows:

```
set samfs:nhino = _value_
```

The range for value is `1` through 1048756. value must be a nonzero power of 2. The default value for `nhino` is one of the following:

- A value that is equal to the ninodes value divided by 8 and then, if necessary, rounded up to the nearest power of 2. For example, assume that the following line exists in `/etc/system`:

  ```
  set samfs:ninodes 8000
  ```

For this example, if `nhino` is not set, the system assumes `1024`, which is 8000 divided by 8 and then rounded up to the nearest power of 2 (210)

- `512`. The file systems set `nhino` to `512` if the `ninodes` setting is out of range.

When to Set the **ninodes** and **nhino** Parameters

When searching for an inode by number (after obtaining an inode number from a directory or after extracting an inode number from an NFS file handle), a Sun QFS file system searches its cache of in-core inodes. To speed this process, the file system maintains a hash table to decrease the number of inodes it must check.

A larger hash table reduces the number of comparisons and searches, at a modest cost in memory usage. If the `nhino` value is too large, the system is slower when undertaking operations that sweep through the entire inode list (inode syncs and unmounts). For sites that manipulate large numbers of files and sites that do extensive amounts of NFS I/O, it can be advantageous to set these parameter values to larger than the defaults.

If your site has file systems that contain only a small number of files, it might be advantageous to make these numbers smaller than the defaults. This could be the case, for example, if you have a file system into which you write large single-file `tar`(1) files to back up other file systems.

Related Topics

- Configuring WORM-FS File Systems
- Using QFS File Systems With SANergy (SAN-QFS)

Using QFS File Systems With SANergy (SAN-QFS)

Contents

Using QFS File Systems with SANergy (SAN-QFS)

Using the SAN-QFS File System in a Heterogeneous Computing Environment

The SAN-QFS file system enables multiple hosts to access the data stored in a Sun QFS system at full disk speeds. This capability can be especially useful for database, data streaming, web page services, or any application that demands high-performance, shared-disk access in a heterogeneous environment.

You can use the SAN-QFS file system in conjunction with fibre-attached devices in a storage area network (SAN). The SAN-QFS file system enables high-speed access to data through Sun QFS software and software such as Tivoli SANergy file-sharing software. To use the SAN-QFS file system, you must have both the SANergy (2.2.4 or later) and the Sun QFS software. For information about the levels of Sun QFS and SANergy software that are supported, contact your Sun sales representative.

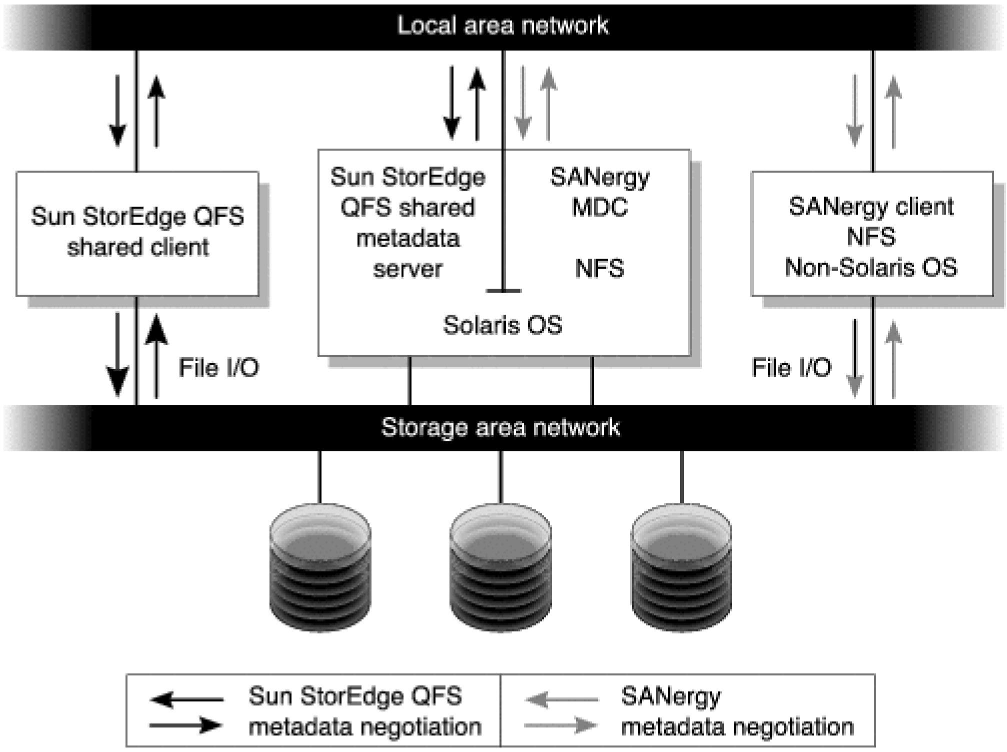
> Note -
> In environments that include the Solaris OS and supported Linux operating systems, use the Sun QFS shared file system, not the SAN-QFS file system, on the Solaris hosts. For information about the Sun QFS shared file system, see Configuring a Shared File System. For a comparison of the Sun QFS shared file system and the SAN-QFS file system, see SAN-QFS Shared File System and Sun QFS Shared File System Comparison.

The following figure depicts a SAN-QFS file system that uses both the Sun QFS software and the SANergy software and shows that the clients and the metadata controller (MDC) system manage metadata across the local area network (LAN). The clients perform I/O directly to and from the storage devices.

Note that all clients running only the Solaris OS are hosting the Sun QFS software, and that all heterogeneous clients running an OS other than Solaris are hosting the SANergy software and the NFS software. The SAN-QFS file system's metadata server hosts both the Sun QFS and the SANergy software. This server acts not only as the metadata server for the file system but also as the SANergy MDC.

> Note -
> The SANergy software is not supported on x64 hardware platforms.

Figure – SAN-QFS File System Using Sun QFS Software and SANergy Software



Enabling the SAN-QFS File System

The following procedures describe how to enable the SAN-QFS file system. Perform these procedures in the order in which they are presented.

Before You Begin

Before you enable the SAN-QFS file system, keep the following configuration considerations in mind and plan accordingly:

- Disks configured for use in a SAN-QFS file system cannot be under the control of a volume manager.
- For the Sun QFS metadata server to be enabled or relocated in a SAN-QFS environment, the new metadata server system must be configured as a SANergy metadata controller.
- A SAN-QFS file system does not recognize segmented files. This can result in unexpected behavior if segmented files are used within the SAN-QFS environment.
- Devices that are classified as `ms` or `md` devices in the Sun QFS `mcf` file are not supported in a SAN-QFS file system.

> Note -
> This documentation assumes that your non-Solaris clients are hosting SANergy software and NFS software for file system sharing. The text and examples in this document reflect this configuration. If your non-Solaris clients host the Samba software instead of the NFS software, see your Samba documentation.

How to Enable the SAN-QFS File System on the Metadata Controller

When you use the SAN-QFS file system, one host system in your environment acts as the SANergy metadata controller (MDC). This is the host system upon which the Sun QFS file system resides.

1. Log in to the host upon which the Sun QFS file system resides and become `superuser`.
2. Verify that the Sun QFS file system is tested and fully operational.
3. Install and configure the SANergy software.
   For instructions, see your SANergy documentation.
4. Use the `pkginfo(1)` command to verify the SANergy software release level:

```
# pkginfo -l SANergy
```

5. Ensure that the file system is mounted.
   Use the mount(1M) command either to verify the mount or to mount the file system.
6. Use the share(1M) command in the following format to enable NFS access to client hosts:

```
MDC# share -F nfs -d _qfs-file-system-name_ _/mount-point_
```

   For qfs-file-system-name, specify the name of your Sun QFS file system, such as qfs1. For more information about the share(1M) command, see the share(1M) or share_nfs(1M) man page.
   For mount-point, specify the mount point of qfs-file-system-name.

7. If you are connecting to Microsoft Windows clients, configure Samba, rather than NFS, to provide security and namespace features.
   To do this, add the SANERGY_SMBPATH environment variable in the /etc/init.d/sanergy file and point it to the location of the Samba configuration file. For example, if your Samba configuration file is named /etc/swf/smb.conf, you must add the following lines to the beginning of your /etc/init.d/sanergy file:
   SANERGY_SMBPATH=/etc/sfw/smb.confexport SANERGY_SMBPATH
8. (Optional) Edit the file system table (/etc/dfs/dfstab) on the MDC to enable access at boot time.
   Perform this step if you want to automatically enable this access at boot time.

How to Enable the SAN-QFS File System on the Clients

After you have enabled the file system on the MDC, you are ready to enable it on the client hosts. The SAN-QFS file system supports several client hosts including IRIX, Microsoft Windows, AIX, and Linux hosts. For information about the specific clients supported, see your Sun sales representative.

Every client has different operational characteristics. This procedure uses general terms to describe the actions you must take to enable the SAN-QFS file system on the clients. For information specific to your clients, see the documentation provided with your client hosts.

Steps

1. Log in to each of the client hosts.
2. Edit the file system defaults table on each client and add the file system.
   For example, on a Solaris OS, edit the /etc/vfstab file on each client and add the name of your Sun QFS file system, as follows:

```
server:/qfs1  -  /qfs1  nfs  -  yes  noac,hard,intr,timeo=1000
```

   On other operating system platforms, the file system defaults table might reside in a file other than /etc/vfstab. For example, on Linux systems, this file is /etc/fstab.
   For more information about editing the /etc/vfstab file, see Sun SAM-QFS Installation and Upgrade Guide. For information about required or suggested NFS client mount options, see your SANergy documentation.

How to Install the SANergy Software on the Clients

After enabling the file system on the client hosts, you are ready to install the SANergy software on the clients. The following procedure describes the SANergy installation process in general terms.

1. Install and configure the SANergy software.
   For instructions, see your SANergy documentation.
2. Use the mount command to NFS mount the file system.
   For example:

```
# mount <host>:/<mount-point>/ <local-mount-point>
```

   For host, specify the MDC.
   For mount-point, specify the mount point of the Sun QFS file system on the MDC.
   For local-mount-point, specify the mount point on the SANergy client.

3. Use the SANergy `fuse` command to fuse the software:

```
#  fuse |<mount-point>
```

For mount-point, specify the mount point on the SANergy client.

Unmounting the SAN-QFS File System

The following procedures describe how to unmount a SAN-QFS file system that is using the SANergy software. Perform these procedures in the order in which they are presented.

How to Unmount the SAN-QFS File System on the SANergy Clients

Follow these steps for each client host on which you want to unmount the SAN-QFS file system.

1. Log in to the client host and become `superuser`.
2. Use the SANergy `unfuse` command to unfuse the file system from the software:

```
# unfuse|<mount-point>
```

For mount-point, specify the mount point on the SANergy client.

3. Use the `umount`(1M) command to unmount the file system from NFS:

```
# umount <host>:/<mount-point>/ <local-mount-point>
```

For host, specify the MDC.
For mount-point, specify the mount point of the Sun QFS file system on the MDC.
For local-mount-point, specify the mount point on the SANergy client.

How to Unmount the SAN-QFS File System on the Metadata Controller

1. Log in to the MDC system and become `superuser`.
2. Use the `unshare`(1M) command to disable NFS access to client hosts:

```
MDC# unshare <qfs-file-system-name> <mount-point>
```

For qfs-file-system-name, specify the name of your Sun QFS file system, such as `qfs1`. For more information about the `unshare(1M)` command, see the share(1M) man page.
For mount-point, specify the mount point of qfs-file-system-name.

How to Unmount the SAN-QFS File System on the Sun QFS Clients

Follow these steps on each participating client host.

1. Log in to a Sun QFS client host and become `superuser`.
2. Use the `umount`(1M) command to unmount the file system.
For example:

```
# umount /qfs1
```

How to Unmount the SAN-QFS File System on the Sun QFS Server

1. Log in to the host system upon which the Sun QFS file system resides and become `superuser`.
2. Use the `umount`(1M) command to unmount the file system.

Troubleshooting: Unmounting a SAN-QFS File System With SANergy File Holds

SANergy software issues holds on Sun QFS files to reserve them temporarily for accelerated access. If SANergy crashes when holds are in effect, you will not be able to unmount the file system. If you are unable to unmount a SAN-QFS file system, examine the `/var/adm/messages` file and look for console messages that describe outstanding SANergy holds.

Whenever possible, allow the SANergy file-sharing function to clean up its holds, but in an emergency, or in case of a SANergy file-sharing system failure, use the following procedure to avoid a reboot.

How to Unmount a File System in the Presence of SANergy File Holds

1. Use the `unshare`(1M) command to disable NFS access.
2. Use the `samunhold`(1M) command to release the SANergy file system holds.
   For more information about this command, see the `samunhold`(1M) man page.
3. Use the `umount`(1M) command to unmount the file system.

Block Quotas in a SAN-QFS File System

The SANergy software does not enforce block quotas. Therefore, it is possible for you to exceed a block quota when writing a file with the SANergy software. For more information on quotas, see Enabling Quotas.

File Data and File Attributes in a SAN-QFS File System

The SANergy software uses the NFS software for metadata operations, which means that the NFS close-to-open consistency model is used for file data and attributes. File data and attributes among SANergy clients do not support the POSIX coherency model for open files.

Using **`samgrowfs`**(1M) to Expand SAN-QFS File Systems

You can use the `samgrowfs`(1M) command to increase the size of a SAN-QFS file system. To perform this task, follow the procedures described in Adding Disk Cache to a File System.

> Caution -
> When using this procedure, be aware that the line-by-line device order in the `mcf` file must match the order of the devices listed in the file system's superblock.

When the `samgrowfs`(1M) command is issued, the devices that were already in the `mcf` file keep their positions in the superblock. New devices are written to subsequent entries in the order in which the are encountered.

If this new order does not match the order in the superblock, the SAN-QFS file system cannot be fused.

SAN-QFS Shared File System and Sun QFS Shared File System Comparison

The SAN-QFS shared file system and the Sun QFS shared file system have the following similarities:

- Both can stage files.
- Both are useful in data capture environments in which it is desirable that the primary file system host not be responsible for writing the data.

- Both are advantageous in environments where there is contention for writing files.

The following table describes differences between the file systems.

Table – SAN-QFS Shared File System Versus Sun QFS Shared File System

SAN-QFS File System	Sun QFS Shared File System
Uses NFS protocol for metadata.	Uses natural metadata.
Preferred in heterogeneous computing environments (that is, when not all hosts are Sun systems).	Preferred in homogeneous Solaris OS environments.
Useful in environments where multiple, heterogeneous hosts must be able to write data.	Preferred when multiple hosts must write to the same file at the same time.

Using the samu Operator Utility

Contents

- File System Commands: Direct I/O Management
- File System Commands: Sun QFS Shared File Systems
- File System Commands: Miscellaneous
- Miscellaneous Commands

Using the **samu**(1M) Operator Utility

This section shows how to use samu(1M) to control the devices configured within your environment. Many samu(1M) displays are useful only for sites using the storage and archive management mechanism. If you are using samu(1M) in a Sun QFS-only environment, these displays do not apply to you.

You can also use the samcmd(1M) command to perform many of the same operations. For more information, see the samcmd(1M) man page.

Operator Utility At a Glance

Option	Description	For More Information
a	Displays archiver status	(a) - Archiver Status Display
c	Displays device configuration	(c) - Device Configuration Display
C	Displays memory information	(C) - Memory Display
d	Displays traced events	(d) - Daemon Trace Controls Display
D	Displays the disk volume dictionary	(D) - Disk Volume Dictionary
f	Displays the components of a Sun QFS file systems	(f) - File Systems Display
F	Displays the label on an optical disk	(F) - Optical Disk Label Display
h	Displays a summary of the available samu(1M) displays	(h) - Help Display
I	Displays the content of inodes	(I) - Inode Display
J	Displays the shared memory segment for the preview queue	(J) - Preview Shared Memory Display
K	Displays kernel statistics	(K) - Kernel Statistics Display
l	Displays usage information for the file system	(l) - Usage Display
L	Displays the location of shared memory tables and system defaults in shared memory	(L) - Shared Memory Tables
m	Displays the status of mass storage mounted file systems and member drives	(m) - Mass Storage Status Display
N	Displays information about the file system	(N) - File System Parameters Display
o	Displays the status of optical disk drives	(o) - Optical Disk Status Display
p	Displays pending load requests for removable media	(p) - Removable Media Load Requests Display
P	Displays the services registered with the Sun QFS single port multiplexer	(P) - Active Services Display
r	Displays the activity of removable media devices	(r) - Removable Media Status Display
R	Displays information and status of Sun SAM-Remote configurations	(R) - Sun SAM-Remote Information Display
s	Displays the status for configured devices	(s) - Device Status Display
S	Displays raw device data	(S) - Sector Data Display
t	Displays the satus of all configured tape drives	(t) - Tape Drive Status Display
T	Displays the SCSI status of a SCSI device	(T) - SCSI Sense Data Display
u	Displays the list of files in the staging queue	(u) - Staging Queue Display

U	Displays the device table in a human-readable form	(U) - Device Table Display
v	Displays the location and VSN of all cataloged disks or tapes	(v) - Automated Library Catalog Display
w	Displays pending stage requests with unloaded volumes	(w) - Pending Stage Queue
down	Terminates operation on device	Device Commands
idle	Restricts access to device by preventing new connections	Device Commands
off	Logically turns off device	Device Commands
on	Logically turns on device	Device Commands
unavail	Makes device unavailable for use with the file system	Device Commands
unload	Unloads the mounted media for the specified removable media device	Device Commands
nalloc	Prohibits any future allocation to the device	Device Commands
alloc	Re-enables allocation to the device	Device Commands
flush_behind	Controls whether modified pages are written to disk asynchronously	The flush_behind Command
force_nfs_async noforce_nfs_async	Controls whether the file system caches NFS data written to the server	The force_nfs_async and noforce_nfs_async Commands
sw_raid nosw_raid	Controls whether the file system aligns the writebehind buffer.	The sw_raid and nosw_raid Commands
readahead	Sets the maximum number of bytes that can be read ahead by the file system	The readahead Command
writebehind	Sets the maximum number of bytes that can be written behind by a file system.	The writebehind Command
wr_throttle	Sets the number of outstanding write kilobytes for one file.	The wr_throttle Command
dio_rd_form_min	Sets the minimum number of blocks for read operations for well-aligned I/O	The dio_rd_form_min and dio_wr_form_min Commands
dio_wr_form_min	Sets the minimum number of blocks for write operations for well-aligned I/O	The dio_rd_form_min and dio_wr_form_min Commands
dio_rd_ill_min	Sets the minimum number of blocks for read operations for misaligned I/O	The dio_rd_ill_min and dio_wr_ill_min Commands
dio_wr_ill_min	Sets the minimum number of blocks for write operations for misaligned I/O	The dio_rd_ill_min and dio_wr_ill_min Commands
dio_rd_consec	Sets the number of consecutive read operations that can occur without regard to a buffer size	The dio_rd_consec and dio_wr_consec Commands
dio_wr_consec	Sets the number of consecutive write operations that can occur without regard to a buffer size	The dio_rd_consec and dio_wr_consec Commands
dio_szero nodio_szero	Controls whether uninitialized areas of sparse files written with direct I/O are zeroed when the area is accessed.	The dio_szero and nodio_szero Commands
forcedirectio noforcedirectio	Controls whether direct I/O or buffered I/O is used as the default mode.	The forcedirectio and noforcedirectio Commands
meta_timeo	Sets the time limit for the Sun QFS shared file system metadata cache	The meta_timeo interval Command
mh_write nomh_write	Controls whether to allow multihost read and write operations	The mh_write and nomh_write Commands
minallocsz	Sets the minimum number of allocated blocks	The minallocsz value and maxallocsz value Commands
maxallocsz	Sets the maximum number of allocated blocks	The minallocsz value and maxallocsz value Commands

rdlease	Sets the amount of time for read leases	The rdlease interval, wrlease interval, and aplease interval Commands
wrlease	Sets the amount of time for write leases	The rdlease interval, wrlease interval, and aplease interval Commands
aplease	Sets the amount of time for append leases	The rdlease interval, wrlease interval, and aplease interval Commands
abr noabr	Controls the application binary recovery (ABR) mount option.	The abr and noabr Commands
dmr nodmr	Controls the direct mirror reads (DMR) mount option.	The dmr nd nodmr Commands
invalid	Sets the number of seconds that the file system holds cached attributes	The invalid interval Command
mm_stripe	Sets the number for disk allocation units for the metadata stripe width	The mm_stripe Command
qwrite noqwrite	Controls whether read and write operations to the same file are performed simultaneously from different threads.	The qwrite and noqwrite Commands
refresh_at_eof norefresh_at_eof	Controls whether Sun QFS hosts update file sizes	The refresh_at_eof and norefresh_at_eof Commands
suid nosuid	Controls whether running programs can change their owner IDs.	The suid and nosuid Commands
stripe	Sets the stripe width for the file system to the specified number of disk allocation units	The stripe Command
sync_meta	Controls whether metadata is written to disk immediately	The sync_meta value Command
trace notrace	Controls whether a file system uses the trace feature	The trace and notrace Commands
clear	Removes the specified volume from the Removable Media Mount Requests display	The clear vsn Command
devlog	Specifies one or more events to be logged	The devlog Command
diskvols	Controls the flags in the disk volume dictionary	The diskvols Command
dtrace	Control the dtrace feature for one or more processes	The dtrace Command
fs	Sets the default file system	The fs Command
mount	Specifies a Sun QFS file system	The mount Command
open	Enables access to a disk device	The open Command
read	Reads the specified sector from an open disk device	The read Command
refresh	Sets the amount of time between samu(1M) screen refreshes	The refresh Command
snap	Sends a copy of an operator display to file	The snap Command
!	Enables you to run a shell command within the samu(1M) operator utility.	The ! shell_command Command

Overview

The samu(1M) operator utility requires a display terminal that displays a minimum of 24 lines by 80 characters wide. The utility includes the following features:

- Displays that enable you to monitor Sun QFS and SAM-QFS devices and file system activity.
- Commands that enable you to select displays, set display options, control access to and the activity of devices, and take snapshots of display windows.
- Commands that enable you to tune a running file system.

The display windows shown in this section are representative examples. The format and amount of information displayed on your terminal can be different, depending on your terminal model and the devices configured in your environment.

The following sections describe how to start and stop samu(1M), interact with the utility, access the help windows, and view operator displays.

How to Invoke **samu**(1M)

To start the operator utility, type the samu(1M) command from the command line:

```
# samu
```

The system starts samu(1M) and shows the help display. This is the default initial display. To view a different samu(1M) display, follow the steps in How to Display a samu(1M) Screen.
To change the default initial display, see the samu(1M) man page.

> Note -
> samu(1M), like the vi(1) editor, is based on the curses(3CURSES) routine. If you have trouble invoking samu(1M), make sure that your terminal type is defined correctly.

How to Display a **samu**(1M) Screen

The samu(1M) command accepts options on its command line for displaying different samu(1M) screens.

1. Type a colon (:) character.
   The following appears in the lower left corner:

   ```
   Command:
   ```

2. Type the letter that corresponds to the display you want to view and press the Return key.
   For example, to view the Automated Library Catalog Display, type a v character and press Return.
   For a complete list of the displays and the letters that correspond to the displays, enter (h) - Help Display.

How to Stop **samu**(1M)

To exit samu(1M), type one of the following:

- q
- :q
  The samu(1M) operator utility exits and returns you to the command shell.

Interacting With **samu**(1M)

The samu(1M) utility is similar the UNIX vi(1) editor with respect to paging forward or backward, entering commands, refreshing the display, and quitting the utility.

For each operator display, the description includes the control key sequences you use to navigate in that display. The samu(1M) man page summarizes the control key sequences.

The last line of the display window shows the error messages. If a command error occurs, automatic display refreshing halts until the next operator action.

Entering a Device

Each device included in the Sun QFS environment is assigned an Equipment Number (for example, 10) in the mcf file. Many samu(1M) commands require you to identify a specific device using its Equipment Number. To see a list of the devices and their Equipment Numbers, see the (c) - Device Configuration Display

Example
The syntax for the :off command is as follows:

```
:off <eq>
```

For eq, type the Equipment Number for the device you are trying to address.

How to Access Online Help

When you start samu(1M), the default display is the first page of the online Help. For more information about the help (h) display, see (h) - Help Display.

1. Type :h
   To move forward or backward from one screen to the next, type the following key sequence:
   - Press Ctrl-f to move to the following page
   - Press Ctrl-b to move backward to previous page.
2. To return to the help display at any time, pressing the h key.

Operator Displays

To view one of the operator displays, press the its corresponding character key. The lowercase characters, a through w, display operational information.

> Note -
> The uppercase samu(1M) displays (C, D, F, I, J, K, L, M, N, P, R, S, T, and U) are designed to be used at a customer site only with the assistance of a member of the technical support staff.
> This topic does not describe these uppercase displays in detail.

When a operator display overflows the available screen area, the word more appears on the bottom of the display indicating additional information, as shown in the following example|#gftiu]. Use Ctrl-f key sequence to page forward and see more content.

Example – **samu**(1M) Screen Indicating More Text Can Be Obtained

```
xb54  54  exb8505  pt03  0  yes  2  0  on
lt55  55  dlt2000  pt02  1  yes  4  0  on  ml65
hp56  56  hpc1716  pt01  1  yes  3  0  on  hp70
hp57  57  hpc1716  pt01  1  yes  4  0  on  hp70
more
```

When samu(1M) prompts you to enter a device, enter its associated Equipment Number. The configuration display (c) shows Equipment Numbers for all removable media devices. To control all displays, use the control keys listed for the display.

The following sections describe the operator displays in alphabetical order. Examples are provided, and when necessary, displays are followed by a table describing the fields displayed.

> Note -
> If you are using only the Sun QFS software, without the archiving features of SAM-QFS, only some of the operator displays will be available.

(**a**) - Archiver Status Display

The a display shows the archiver status.

- To show the status of the archiver for each file-system, type the command with the following format:

```
Command:a
```

- To display archiving details for a specific file system, type the command with the following format:

```
Command:a <filesystem>
```

For filesystem, specify the name of a file system.

Navigation

The following table shows the control keys you can use in the a display.

Control Keys for the **a** Display

Key	Function
Ctrl-b	Previous file system
Ctrl-f	Next file system
Ctrl-d	Page arcopies forward (bottom portion)
Ctrl-u	Page arcopies backward (bottom portion)

The following table shows the control keys you can use in the `:a <filesystem>` display.

Control Keys for the **`:a <filesystem>`** Display

Key	Function
Ctrl-b	Previous file system
Ctrl-f	Next file system

Sample Display

This example shows activity and statistics for a single file system in the summary display.

Example: **samu**(1M) **a** Display

```
Archiver Status
samu 5.0 07:44:02 August 8 2008
sam-archiverd:  Waiting for resources
sam-arfind:  samfs1 mounted at /sam1
Waiting until 2005-05-08 07:54:02 to scan .inodes
sam-arfind:  samfs2 mounted at /sam2
Waiting until 2005-05-08 07:52:57 to scan .inodes
sam-arfind:  qfs1 mounted at /qfs1
Waiting until 2005-05-08 07:44:33 to scan .inodes
sam-arfind:  qfs2 mounted at /qfs2
Waiting until 2005-05-08 07:53:21 to scan .inodes
sam-arfind:  qfs3 mounted at /qfs3
Waiting until 2005-05-08 07:44:11 to scan .inodes

sam-arfind:  qfs4 mounted at /qfs4
Waiting until 2005-05-08 07:53:35 to scan .inodes

sam-arfind:  shareqfs1 mounted at /shareqfs1
Shared file system client.  Cannot archive.

sam-arfind:  shareqfs2 mounted at /shareqfs2
Shared file system client.  Cannot archive.

sam-arcopy:  qfs4.arset5.1.83 dt.DAT001
Waiting for volume dt.DAT001
```

Field Descriptions

The following table shows the fields in the detail display.

samu a Display Field Descriptions

Field	Description
`samfs1 mounted at`	Mount point.
`regular files`	Number of regular files and their total size.
`offline files`	Number of offline files and their total size.
`archdone files`	Number of `archdone` files and size. Indicates that the archiver has completed processing and can perform no further processing. Files marked as `archdone` have been processed for archiving but have not necessarily been archived.
`copy1`	Number of files and total size for archive copy 1.
`copy2`	Number of files and total size for archive copy 2.
`copy3`	Number of files and total size for archive copy 3.
`copy4`	Number of files and total size for archive copy 4.
`Directories`	Number of directories and total size.
`sleeping until`	Indicates when archiver runs again.

(**c**) - Device Configuration Display

The c display shows your configuration's connectivity. It lists all device names and Equipment Numbers. To invoke the device configuration display, type the command with the following format:

```
Command:c
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **c** Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-u	Half-page backward

Sample Display

This example shows the device configuration display.

Example:**samu c** Display

```
Device configuration:                  samu     5.0 07:48:11 Sept 8 2008
ty  eq state   device_name                     fs family_set
sk  100 on      /etc/opt/SUNWsamfs/dcstkconf    100 dcL700
tp  120 off     /dev/rmt/1cbn                   100 dcL700
sg  130 on      /dev/rmt/4cbn                   100 dcL700
sg  140 on      /dev/rmt/5cbn                   100 dcL700
tp  150 off     /dev/rmt/3cbn                   100 dcL700
hy  151 on      historian                       151
```

Field Descriptions

The following table shows the field descriptions for this display.

samu c Display Field Descriptions

Field	Description
`ty`	Device type.
`eq`	Equipment Number of the device.
`state`	Current operating state of the device. Valid device states are as follows: • on The device is available for access. • `ro` The device is available for read-only access. • `off` The device is not available for access. • down The device is available only for maintenance access. • `idle` The device is not available for new connections. Operations in progress continue until completion. • `nalloc` The `nalloc` flag has been set, which prohibits any future allocation to this device. For more information, see Per-logical unit number (LUN) Allocation Control.
`device_name`	Path to the device.
`fs`	Family Set Equipment Number.
`family_set`	Name of the storage Family Set or library to which the device belongs.

(**C**) - Memory Display

The C display shows the content of a specific memory address. To show the content at an address, enter the address in hexadecimal.
To invoke this display, type the following command:

```
Command:C <hex-address>
```

For hex-address, specify the address of a memory location in hexadecimal. For example:

```
Command:C 0x1044a998
```

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This example shows the memory display. The sample output has been truncated.

Example:**samu C** Display

```
Memory    base: 0x1234567               samu 5.0 07:52:25 Sept 8 2008
00000000  80921000 137ff801 edd05e21 40853420  ......x.mP^!@.4
00000010  00a00420 018528b0 01a604e0 02840640  . . ..(0.&amp;.&rdquo;...@
00000020  02d030a1 a0853420 0080a0a0 100a6fff  .P0! .4 ..  ..o.
00000030  f6921000 13c65e23 582d0000 0ba01020  v....F^#X-... .
00000040  00c45e20 48c608e0 2fd05e21 40920080  .D^ HF.&rdquo;/P^!@...
00000050  037ff801 fa941000 16c45e20 48a600a0  ..x.z....D^ H&amp;.
00000060  80921000 137ff801 d5d05e21 40853420  ......x.UP^!@.4
00000070  00a00420 018528b0 01a604e0 02840640  . . ..(0.&amp;.&rdquo;...@
00000080  02d030a1 c0853420 0080a0a0 100a6fff  .P0!@.4 ..  ..o.
00000090  f6921000 13c65e23 58a01020 00c45e20  v....F^#X . .D^
000000a0  48c608e0 2fd05e21 40920080 037ff801  HF.&rdquo;/P^!@.....x.
000000b0  e39405a2 00c45e20 48a600a0 80921000  c..".D^ H&amp;. ....
000000c0  137ff801 bed05e21 40853420 00a00420  ..x.>P^!@.4 . .
000000d0  018528b0 01a604e0 02840640 02d030a1  ..(0.&amp;.&rdquo;...@.P0!
000000e0  e0853420 0080a0a0 100a6fff f6921000  &rdquo;.4 ..  ..o.v...
000000f0  13c65e23 58a01020 00c45e20 48c608e0  .F^#X . .D^ HF.&rdquo;
```

(**d**) - Daemon Trace Controls Display

The d display shows the events being traced as specified in the `defaults.conf` file. For more information about enabling trace files, see the `defaults.conf`(4) man page.
To invoke this display, type the following command:

```
Command:d
```

Sample Display

This example shows trace file information. It includes information about the daemons being traced, the paths to the trace files, the events being traced, and information about the size and age of the trace files.

Example:**samu d** Display

```
Daemon trace controls                   samu 5.0 07:56:38 Sept 8 2008
sam-amld       /var/opt/SUNWsamfs/trace/sam-amld
               cust err fatal misc proc debug date
               size    0    age 0
sam-archiverd  /var/opt/SUNWsamfs/trace/sam-archiverd
               cust err fatal misc proc debug date
               size    0    age 0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
               cust err fatal misc proc debug date
               size    0    age 0
sam-fsd        /var/opt/SUNWsamfs/trace/sam-fsd
               cust err fatal misc proc debug date
               size    0    age 0
sam-rftd       /var/opt/SUNWsamfs/trace/sam-rftd
               cust err fatal misc proc debug date
               size    0    age 0
sam-recycler   /var/opt/SUNWsamfs/trace/sam-recycler
               cust err fatal misc proc debug date
               size    0    age 0
sam-sharefsd   /var/opt/SUNWsamfs/trace/sam-sharefsd
               cust err fatal misc proc debug date
               size    0    age 0
sam-stagerd    /var/opt/SUNWsamfs/trace/sam-stagerd
               cust err fatal misc proc debug date
               size    0    age 0
sam-serverd    /var/opt/SUNWsamfs/trace/sam-serverd
               cust err fatal misc proc debug date
               size    0    age 0
sam-clientd    /var/opt/SUNWsamfs/trace/sam-clientd
               cust err fatal misc proc debug date
               size    0    age 0
sam-mgmt       /var/opt/SUNWsamfs/trace/sam-mgmt
               cust err fatal misc proc debug date
               size    0    age 0
```

(D) - Disk Volume Dictionary

The D display shows the disk volume dictionary, which keeps track of the disk media for disk archiving that has been defined in the `diskvols.conf` file. The dictionary contains information about each VSN, including the capacity, space remaining, and the status flags of the VSN. These flags include unavailable, read only, and bad media.
To invoke this display, type the following command:

```
Command:D
```

Sample Display

This example shows the device configuration display.

Example:**samu D** Display

```
Disk volume dictionary     samu     5.0 07:48:11 May 8 2008

header
version 460

volumes
magic 340322 version 9 nkeys 2 ndata 2
index    space          capacity       used        flags    volume
    0    12882411520    12887785472    10291200    -----    disk01
    1     6443827200     6443892736       70656    -----    disk02
clients
magic 340322 version 9 nkeys 1 ndata 1
```

Flags

The following table shows the flags for the D display.

Flag Values for the **samu D** Display

Flag	Description
l	Volume is labeled; seqnum file has been created. This is set by the administrator to prevent the software from creating a new seqnum file.
r	Volume is defined on a remote host.
U	Volume is unavailable.
R	Volume is read only.
E	Media error, indicating the software detects a write error on the disk archive directory.

To set or clear a disk volume dictionary flag, use the diskvols samu(1M) command. See The diskvols Command.

(**f**) - File Systems Display

The f display shows the components of your Sun QFS file systems.
To invoke this display, type the following command:

```
Command:f
```

Sample Display

This example shows the file systems display.

Example:**samu f** Display

```
File systems                                     samu 5.0 08:11:24 Sept 8 2008

ty      eq state          device_name   status     high low mountpoint server
ms      10    on               samfs1  m----2----d 90% 70% /sam1
 md     11    on     /dev/dsk/c5t8d0s3
 md     12    on     /dev/dsk/c5t8d0s4
 md     13    on     /dev/dsk/c5t8d0s5
 md     14    on     /dev/dsk/c5t8d0s6
 md     15    on     /dev/dsk/c5t8d0s7
ms      20    on               samfs2  m----2----d 90% 70% /sam2
 md     21    on     /dev/dsk/c5t9d0s3
 md     22    on     /dev/dsk/c5t9d0s4
 md     23    on     /dev/dsk/c5t9d0s5
 md     24    on     /dev/dsk/c5t9d0s6
 md     25    on     /dev/dsk/c5t9d0s7
ma      30    on                 qfs1  m----2----d 90% 70% /qfs1
 mm     31    on    /dev/dsk/c5t10d0s0
 md     32    on    /dev/dsk/c5t10d0s1
ma      40    on                 qfs2  m----2----d 90% 70% /qfs2
 mm     41    on    /dev/dsk/c5t11d0s0
 md     42    on    /dev/dsk/c5t11d0s1
ma      50    on                 qfs3  m----2---r- 90% 70% /qfs3
 mm     51    on    /dev/dsk/c5t12d0s0
 mr     52    on    /dev/dsk/c5t12d0s1
ma      60    on                 qfs4  m----2---r- 90% 70% /qfs4
 mm     61    on    /dev/dsk/c5t13d0s0
 mr     62    on    /dev/dsk/c5t13d0s1
ma     100    on            shareqfs1  m----2c--r- 80% 70% /shareqfs1 spade
 mm    101    on /dev/dsk/c6t50020F2300004655d0s0
 mr    102    on /dev/dsk/c6t50020F2300004655d0s1
ma     110    on            shareqfs2  m----2c--r- 80% 70% /shareqfs2 spade
 mm    111    on /dev/dsk/c6t50020F2300004655d0s6
 mr    112    on /dev/dsk/c6t50020F2300004655d0s7
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) **f** Display Field Descriptions

Field	Description
`ty`	Device type.
`eq`	Equipment Number of the device.
`state`	Current operating state of the device. Valid device states are as follows: • `on` The device is available for access. • `ro` The device is available for read-only access. • `off` The device is not available for access. • `down` The device is available only for maintenance access. • `idle` The device is not available for new operations. Operations in progress continue until completion. • `nalloc` The `nalloc` flag has been set, which prohibits any future allocation to this device. For more information, see Per-logical unit number (LUN) Allocation Control.
`device_name`	File system name or path to the device.
`status`	Device status. For a description of status codes, see Operator Display Status Codes.
`high`	High disk usage threshold percentage.
`low`	Low disk usage threshold percentage.
`mountpoint`	Mount point of the file system.
`server`	Name of the host system upon which the file system is mounted.

(**F**) - Optical Disk Label Display

The F display shows the label on an optical disk.
To invoke this display, type the following command:

```
Command:F
```

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

(**h**) - Help Display

The h display shows a summary of the `samu`(1M) displays available. By default, this is the first display that the system presents when you enter the `samu`(1M) command at the command line.
To invoke this display, type the following command:

```
Command:h
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **h** Display

Key	Function
Ctrl -b	Page backward
Ctrl -d	Page forward (top portion)
Ctrl -f	Page forward
Ctrl -u	Page backward (top portion)
Ctrl -k	Toggle path display

Sample Display

The online Help has several pages of help screens, but this example shows only the first page of the SAM-QFS configuration. Subsequent help screens show `samu`(1M) commands.
On a Sun QFS file system, not all of the displays appear in the initial help screen. For example, the removable media displays are not available if you are running a Sun QFS system.

Example: **`samu`**(1M) Initial Help Screen for a Sun SAM System

```
Help information    page 1/15    samu 5.0     08:18:13 Sept 8 2008
Displays:
    a    Archiver status                    w    Pending stage queue
    c    Device configuration               C    Memory
    d    Daemon trace controls              D    Disk volume dictionary
    f    File systems                       F    Optical disk label
    h    Help information                   I    Inode
    l    Usage information                  J    Preview shared memory
    m    Mass storage status                K    Kernel statistics
    n    Staging status                     L    Shared memory tables
    o    Optical disk status                M    Shared memory
    p    Removable media load requests      N    File system parameters
    r    Removable media                    P    Active Services
    s    Device status                      R    SAM-Remote
    t    Tape drive status                  S    Sector data
    u    Staging queue                      T    SCSI sense data
    v    Robot catalog                      U    Device table

    more (ctrl-f)
```

(**I**) - Inode Display

The `I` display shows the content of inodes.
You can invoke this display differently, depending on what you need to view, as follows:

- To display inodes for an entire file system, type the command with the following format:

```
Command:I <filesystem>
```

For filesystem, specify the name of a file system.

- To display a specific inode, type the command with the following format:

```
Command:I <inode-number>
```

For inode-number, specify the inode number in either hexadecimal or decimal.

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **I** Display

Key	Function
Ctrl-b	Previous inode
Ctrl-f	Next inode
Ctrl-k	Advance display format

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This example shows the inode display.

Example:**samu**(1M) **I** Display

```
Inode     0x1 (1) format: file          samu     5.0 08:27:14 Sept 8 2008
    incore: y

00008100 mode       -r--------          409cdf57 access_time
00000001 ino        (1)                 1d32ea20
00000001 gen        (1)                 4096b499 modify_time
00000002 parent.ino (2)                 02588660
00000002 parent.gen (2)                 4096b499 change_time
00000000 size_u                         02588660
000c0000 size_l     (786432)            4096b443 creation_time
01000000 rm:media/flags                 409a8a7c attribute_time
00000000 rm:file_offset                 409c0ce6 residence_time
00000000 rm:mau                         00000000 unit/cs/arch/flg
00000000 rm:position                    00000000 ar_flags
00000000 ext_attrs  --------            00000000 stripe/stride/sg
00000000 ext.ino    (0)                 00000000 media  -- --
00000000 ext.gen    (0)                 00000000 media  -- --
00000000 uid        root                00000000 psize      (0)
00000000 gid        root                000000c0 blocks     (192)
00000001 nlink      (1)                 00000600 free_ino   (1536)
00011840 status -n-------  ----- -- --

Extents (4k displayed as 1k):
00_ 000000d0.00 000000e0.00 000000f0.00 00000100.00 00000110.00 00000120.00
06_ 00000130.00 00000140.00 00000150.00 00000160.00 00000170.00 00000180.00
12_ 00000190.00 000001a0.00 000001b0.00 000001c0.00 00000630.00 00000000.00
18_ 00000000.00
```

(**J**) - Preview Shared Memory Display

The J display shows the shared memory segment for the preview queue.
To invoke this display, type the following command:

```
Command:J
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **J** Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-u	Half-page backward

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This example shows the preview shared memory display. This sample output has been truncated.

Example: **samu**(1M) **J** Display

```
Preview shared memory   size: 155648    samu 5.0 08:30:05 Sept 8 2008

00000000  00040000 00014d58 00000000 00000000  ......MX........
00000010  00000000 00000000 73616d66 73202d20  ........samfs -
00000020  70726576 69657720 6d656d6f 72792073  preview memory s
00000030  65676d65 6e740000 00026000 00000000  egment..........
00000040  00025fff 00000000 00040000 00014d58  .._..........MX
00000050  00000000 00000000 00000000 00000000  ................
00000060  0000d9e0 00000064 00000000 000001b8  ..Y.....d.......8
00000070  3f800000 447a0000 0000d820 00000008  ?...Dz....X ....
```

(**K**) - Kernel Statistics Display

The K display shows kernel statistics, such as the number of inodes currently in memory.
To invoke this display, type the following command:

```
Command:K
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **K** Display

Key	Function
Ctrl-b	Page backward
Ctrl-f	Page forward

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This example shows the kernel statistics display.

Example: **samu**(1M) **K** Display

```
Kernel statistics                       samu 5.0     08:33:19 Sept 8 2008

module: sam-qfs  name: general instance: 0 class: fs
version                       4.4.sam-qfs, gumball 2004-05-07 12:12:04
configured file systems       8
mounted file systems          8
nhino                         16384
ninodes                       129526
inocount                      129527
inofree                       128577
```

(**l**) - Usage Display

The `l` display shows the usage information for the file system, including the capacity and space used for each library and file system.
To invoke this display, type the following command:

```
Command:l
```

Sample Display

This example shows an example of a usage display.

Example:**samu**(1M) **l** Display

```
Usage information                     samu     5.0 08:36:27 Sept 8 2008

hostid = 80e69e6e OS name: SunOS Architecture: SPARC CPUs: 2 (2 online)

library     40: capacity    389.3G     bytes space    291.1G    bytes, usage     25%
library     51: capacity    9.5G       bytes space    9.5G      bytes, usage     0%
library     55: capacity    0          bytes space    0         bytes, usage     0%
library     56: capacity    10.7G      bytes space    10.7G     bytes, usage     0%
library totals: capacity    409.5G     bytes space    311.3G    bytes, usage     24%

filesystem samfs3: capacity    54.5M    bytes space    13.4M    bytes, usage     75%
filesystem samfs4: capacity    319.5M   bytes space    298.0M   bytes, usage      7%
filesystem samfs7: capacity    96.6M    bytes space    69.6M    bytes, usage     28%
filesystem samfs6: capacity    5.0G     bytes space    4.9G     bytes, usage      3%
filesystem samfs8: capacity    5.0G     bytes space    4.9G     bytes, usage      2%
filesystem totals: capacity    10.5G    bytes space    10.2G    bytes, usage      3%
```

Note -
In versions of the software before 4.3, this display showed license information for the file system.

(**L**) - Shared Memory Tables

The L display shows the location of the shared memory tables. It also shows some system defaults that are kept in shared memory.
To invoke this display, type the following command:

```
Command:L
```

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This shows the shared memory tables.

Example: **samu**(1M) **L** Display

```
Shared memory tables                    samu 5.0 08:38:31 May  8 2008

shm ptr tbl:                            defaults:
size            12000 (73728)           optical         mo
left            44c8 (17608)            tape            lt
scanner pid     1861                    timeout         600
fifo path       01b0 /var/opt/SUNWsamfs/previews        100
dev_table       01cc                    stages          1000
first_dev       0450                    log_facility    184
scan_mess       cf50                    dio minfilesize 100
preview_shmid   1                       label barcode   FALSE
flags           0x20000000              barcodes low    FALSE
preview stages  55776                   export unavail  FALSE
preview avail   100                     attended        TRUE
preview count   0                       start rpc       FALSE
preview sequence  445
age factor         1                    vsn factor      1000
fs tbl ptr 0xd820                       fs count  8
fseq  10 samfs1 state 0      0      0      0      0
fseq  20 samfs2 state 0      0      0      0      0
fseq  30 qfs1 state 0      0      0      0      0
fseq  40 qfs2 state 0      0      0      0      0
fseq  50 qfs3 state 0      0      0      0      0
fseq  60 qfs4 state 0      0      0      0      0
fseq  100 shareqfs1 state 0      0      0      0      0
fseq  110 shareqfs2 state 0      0      0      0      0
```

(m) - Mass Storage Status Display

The m display shows the status of mass storage file systems and their member drives. This display shows only mounted file systems.
To invoke this display, type the following command:

```
Command:m
```

Sample Display

This example shows the m display. Member drives are indented one space and appear directly below the file system to which they belong.

Example: **samu**(1M) **m** Display

```
Mass storage status                      samu 5.0     08:41:11 Sept 8 2008

ty      eq  status       use state ord  capacity      free    ra   part high low
ms      10  m----2----d   1% on          68.354G   68.343G    1M     16  90% 70%
 md     11                1% on      0   13.669G   13.666G
 md     12                1% on      1   13.669G   13.667G
 md     13                1% on      2   13.669G   13.667G
 md     14                1% on      3   13.674G   13.672G
 md     15                1% on      4   13.674G   13.672G
ms      20  m----2----d   1% on          68.354G   68.344G    1M     16  90% 70%
 md     21                1% on      0   13.669G   13.667G
 md     22                1% on      1   13.669G   13.667G
 md     23                1% on      2   13.669G   13.667G
 md     24                1% on      3   13.674G   13.672G
 md     25                1% on      4   13.674G   13.672G
ma      30  m----2----d   4% on          64.351G   61.917G    1M     16  90% 70%
 mm     31                1% on      0    4.003G    3.988G [8363840 inodes]
 md     32                4% on      1   64.351G   61.917G
ma      40  m----2----d   1% on          64.351G   64.333G    1M     16  90% 70%
 mm     41                1% on      0    4.003G    3.997G [8382784 inodes]
 md     42                1% on      1   64.351G   64.333G
ma      50  m----2---r-   1% on          64.351G   64.333G    1M     16  90% 70%
 mm     51                1% on      0    4.003G    3.997G [8382784 inodes]
 mr     52                1% on      1   64.351G   64.333G
ma      60  m----2---r-   1% on          64.351G   64.331G    1M     16  90% 70%
 mm     61                1% on      0    4.003G    3.997G [8382784 inodes]
 mr     62                1% on      1   64.351G   64.331G
ma     100  m----2c--r-   2% on         270.672G  265.105G    1M     16  80% 70%
 mm    101                1% on      0    2.000G    1.988G [4168992 inodes]
 mr    102                2% on      1  270.672G  265.469G
ma     110  m----2c--r-   3% on         270.656G  263.382G    1M     16  80% 70%
 mm    111                1% on      0    2.000G    1.987G [4167616 inodes]
 mr    112                2% on      1  270.656G  264.736G
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) **m** Display Field Descriptions

Field	Description
ty	Device type.
eq	Equipment Number of the mass storage device.
status	Device status. For a description of status codes, see Operator Display Status Codes.
use	Percentage of disk space in use.
state	Current operating state of the mass storage device.
ord	Ordinal number of the disk device within the storage Family Set.
capacity	Number of 1024-byte blocks of usable space on the disk.
free	Number of 1024-byte blocks of disk space available.
ra	Read-ahead size in kilobytes.
part	Partial stage size in kilobytes.
high	High disk usage threshold percentage.
low	Low disk usage threshold percentage.

(**M**) - Shared Memory Display

The M display shows the raw shared memory segment in hexadecimal. This is a device table.
To invoke this display, type the following command:

```
Command:M
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **M** Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-u	Half-page backward

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This example shows the shared memory display. The sample output has been truncated.

Example:**samu**(1M) **M** Display

```
Shared memory       size: 73728         samu 5.0     08:43:20 May  8 2008

00000000  00040000 00014d58 00000000 00000000  ......MX........
00000010  00000000 00000000 73616d66 73202d20  ........samfs -
00000020  73686172 6564206d 656d6f72 79207365  shared memory se
00000030  676d656e 74000000 00012000 000044c8  gment..... ...DH
00000040  0000dd20 00000000 00000742 00000745  ..] .......B...E
00000050  00000001 00000000 00000000 c0000000  ............@...
00000060  00000001 0001534d 00000000 00000000  ......SM........
00000070  00000000 00000000 00000000 00000000  ................

00000080  00000000 00000000 00000000 00000000  ................
00000090  20000000 000001b0 000001cc 00000450   ......0...L...P
000000a0  0000cf50 00000001 00000001 4c696365  ..OP........Lice
000000b0  6e73653a 204c6963 656e7365 206e6576  nse: License nev
000000c0  65722065 78706972 65732e00 00000000  er expires......
000000d0  00000000 00000000 00000000 00000000  ................
000000e0  00000000 00000000 00000000 00000000  ................
000000f0  00000000 00000000 00000000 00000000  ................
```

(**n**) - Staging Status Display

The n display shows the status of the stager for all media. It displays a list of outstanding stage requests.

- To display the staging status for all staging activity, type the command with the following format:

```
Command:n
```

- To display the staging status for a specific media type, type the command with the following format:

```
Command:n <mt>
```

For mt, specify one of the media types shown in the `mcf`(4) man page.

Sample Display

This example shows the staging status display.

Example:**samu**(1M) **n** Display

```
Staging status                          samu 5.0     08:47:16 May  8 2008

Log output to: /var/opt/SUNWsamfs/stager/log

Stage request: dt.DAT001
Loading VSN DAT001

Staging queues
ty pid    user          status     wait  files  vsn
dt 16097  root          active     0:00    12   DAT001
```

(**N**) - File System Parameters Display

The N display shows all mount point parameters, the superblock version, and other file system information.
To invoke this display, type the following command:

```
Command:N
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **N** Display

Key	Function
Ctrl -b	Previous file system
Ctrl -d	Page partitions forward
Ctrl -f	Next file system
Ctrl -i	Detailed status interpretations
Ctrl -u	Page partitions backward

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This example shows the file system parameters display.

Example:**samu**(1M) **N** Display

```
File system parameters                   samu 5.0     08:55:19 Sept 8 2008

mount_point     : /sam1                  partial         : 16k
fs_type         : 6                      maxpartial      : 16k
server          :                        partial_stage   : 16384
filesystem name: samfs1                  flush_behind    : 0
eq  type        : 10 ms                  stage_flush_beh: 0
state version   :    0    2              stage_n_window : 262144
(fs,mm)_count   :    5    0              stage_retries  : 3
sync_meta       : 0                      stage timeout  : 0
stripe          : 0                      dio_consec r,w :    0    0
mm_stripe       : 1                      dio_frm_min r,w:  256  256
high  low       :  90%  70%              dio_ill_min r,w:    0    0
readahead       : 1048576                ext_bsize      : 4096
writebehind     : 524288
wr_throttle     : 16777216
rd_ino_buf_size: 16384
wr_ino_buf_size: 512
config          : 0x08520530             mflag          : 0x00000044
status          : 0x00000001

Device configuration:
ty   eq state   device_name                    fs family_set
md   11 on      /dev/dsk/c5t8d0s3              10 samfs1
md   12 on      /dev/dsk/c5t8d0s4              10 samfs1
md   13 on      /dev/dsk/c5t8d0s5              10 samfs1
md   14 on      /dev/dsk/c5t8d0s6              10 samfs1
md   15 on      /dev/dsk/c5t8d0s7              10 samfs1
```

(**o**) - Optical Disk Status Display

The o display shows the status of all optical disk drives configured within the environment.
To invoke this display, type the following command:

```
Command:o
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **o** Display

Key	Function
Ctrl-b	Page backward
Ctrl-f	Page forward
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-k	Select (manual, automated library, both, priority)
Ctrl-u	Half-page backward

Sample Display

This example shows the optical disk status display.

Example: **samu**(1M) **o** Display

```
Optical disk status            samu    5.0     Thu Oct 11 13:15:40
ty  eq  status      act  use  state  vsn
mo  35  --l---wo-r    1  29%  ready  oper2
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) **o** Display Field Descriptions

Field	Description
`ty`	Device type.
`eq`	Equipment Number of the optical disk.
`status`	Device status. For a description of status codes, see Operator Display Status Codes.
`act`	Activity count.
`use`	Percentage of cartridge space used.
`state`	Current operating state of the optical disk. Valid device states are as follows: • `ready` The device is on, and the disk is loaded in the transport; available for access. • `notrdy` The device is on, but no disk is present in the transport. • `idle` The device is not available for new connections. Operations in progress continue until completion. • `off` The device is not available for access. • down The device is available only for maintenance access.
`vsn`	Volume Serial Name assigned to the optical disk, or the keyword `nolabel` if the volume is not labeled.

(**p**) - Removable Media Load Requests Display

The p display lists information about pending load requests for removable media. You can use the mt argument to select either a specific type of media, such as DLT tape, or a family of media, such as tape. The priority display lists the priority in the preview queue, rather than the user queue, and sorts the entries by priority.
It displays mount requests in the following formats:

- Both manual and automated library requests by user
- Both manual and automated library requests by priority
- Manual requests only
- Automated library requests only

You can invoke this display differently, depending on what you need to view, as follows:

- To display mount requests for all removable devices currently selected, type the command with the following format:

  ```
  Command:p
  ```

- To display mount requests for devices of a given removable media type, type the command with the following format:

  ```
  Command:p <mt>
  ```

  For mt, specify one of the media types shown in the `mcf`(4) man page.

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **p** Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-k	Toggle between the different display formats
Ctrl-u	Half-page backward

Sample Display

This example shows the removable media load requests display.

Example: **samu**(1M) **p** Display

```
Removable media load requests all both  samu 5.0 09:14:19 Sept 8 2008
    count: 1

index type pid     user         rb   flags     wait count  vsn
   0  dt   15533   root         150  W--f---   0:00        DAT001
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) **p** Display Field Descriptions

Field	Description
`index`	Index number in the preview table.
`type`	Device type code assigned to the removable media.
`pid`	UNIX process identifier. A process identifier of 1 indicates NFS access.
`user`	Name assigned to the user requesting the load.
`priority`	Priority of the request.
`rb`	Equipment Number of the automated library in which the requested VSN resides.
`flags`	Flags for the device. See the Flag table.
`wait`	The elapsed time since the mount request was received.
`count`	The number of requests for this VSN, if it is a stage.
`vsn`	Volume serial name of the volume.

Flags

The following table shows the flags for the p display.

Flags Field for the **samu**(1M) **p** Display

Flag	Description

W------	Write access requested.
}}{{{}b----	Entry is busy.
-C---	Clear VSN requested.
--f--	File system requested.
---N-	Media is foreign to the file system.
---S{-}	Flip side already mounted.
------s	Stage request flag.

(**P**) - Active Services Display

The P display lists the services registered with the Sun QFS single port multiplexer.
To invoke this display, type the following command:

```
Command:P
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **P** Display

Key	Function
Ctrl-b	Page backward
Ctrl-f	Page forward

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This example shows the active services display.

Example: **samu**(1M) **P** Display

```
Active Services                          samu     5.0    09:08:33 Sept 8 2008

Registered services for host &rdquo;pup&rsquo;:
    sharedfs.qfs2
    sharedfs.qfs1
  2 services registered.
```

(**r**) - Removable Media Status Display

The r display monitors the activity on removable media devices such as tape drives. You can monitor either a specific type of device, such as video tape, or a family of devices such as all tape devices.

- To display the status for all removable media devices, type the command with the following format:

  ```
  Command:r
  ```

- To display the status for a specific device, type the command with the following format:

```
Command:r <eq>
```

For eq, specify the Equipment Number for the device.

Sample Display

This example shows the removable media status display.

Example: **samu**(1M) **r** Display

```
Removable media status: all             samu 5.0     09:11:27 Sept 8 2008

ty   eq  status      act  use  state     vsn
dt  150  --l------r    0  63%  ready     DAT001
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) **r** Display Field Descriptions

Field	Description
`ty`	Device type.
`eq`	Equipment Number of the drive.
`status`	Device status. For a description of status codes, see Operator Display Status Codes.
`act`	Activity count.
`use`	Percentage of cartridge space used.
`state`	Current operating state of the removable media. Valid device states are as follows: • `ready` The device is on, and the disk or tape is loaded in the transport; available for access. • `notrdy` The device is on, but no disk or tape is present in the transport. • `idle` The device is not available for new connections. Operations in progress continue until completion. • `off` The device is not available for access. • down The device is available only for maintenance access. • `nalloc` The `nalloc` flag has been set, which prohibits any future allocation to this device. For more information, see Per-logical unit number (LUN) Allocation Control.
`vsn`	Volume serial name assigned to the volume, or the keyword `nolabel` if the volume is not labeled. Blank if no volume is present in the transport, or device is off.

(**R**) - Sun SAM-Remote Information Display

The R display shows information and status on Sun SAM-Remote configurations.
To invoke this display, type the following command:

```
Command:R
```

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

(**s**) - Device Status Display

The `s` display shows the status for all devices configured within the environment.

To invoke this display, type the following command:

```
Command:s
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **s** Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-u	Half-page backward

Sample Display

This example shows the device status display.

Example: **samu**(1M) **s** Display

```
Device status                           samu     5.0     09:14:05 Sept 8 2008

ty     eq     state    device_name                       fs  status        pos
sk    100     on       /etc/opt/SUNWsamfs/dcstkconf      100 m--------r    stk_dismount(2275) 0,
volser 700073
sg    120     on       /dev/rmt/2cbn                     100 ---------p    empty
sg    130     on       /dev/rmt/5cbn                     100 --l----o-r    Ready for data transfer
sg    140     on       /dev/rmt/6cbn                     100 ---------p    empty
sg    150     on       /dev/rmt/4cbn                     100 ---------p    empty
hy    151     on       historian                         151 ----------
```

Field Descriptions

The following table shows the field descriptions for this display.

{samu}}(1M) **s** Display Field Descriptions

Field	Description
`ty`	Device type.
`eq`	Equipment Number of the device.
`state`	Current operating state of the device.
`device_name`	Path to the device. For file system devices, this is the file system name.
`fs`	Equipment Number of the family, set to which the device belongs.
`status`	Device status. For a description of status codes, see Operator Display Status Codes.

(**S**) - Sector Data Display

The S display shows raw device data.

To invoke this display, type the following command:

```
Command:S
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **S** Display

Key	Function
Ctrl-b	Previous sector
Ctrl-d	Page forward (top portion)
Ctrl-f	Next sector
Ctrl-k	Advance display format
Ctrl-u	Page backward (top portion)

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

(**t**) - Tape Drive Status Display

The `t` display shows the status of all tape drives configured within the environment.
To invoke this display, type the following command:

```
Command:t
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **t** Display

Key	Function
Ctrl-b	Page backward
Ctrl-f	Page forward

Sample Display

This example shows the tape drive status display.

Example:**samu**(1M) **t** Display

```
Tape drive status                      samu     5.0     09:21:07 Sept 8 2008

ty   eq  status      act  use  state    vsn
sg  120  ---------p    0   0%  notrdy          empty
sg  130  ---------p    0   0%  notrdy          empty
sg  140  ---------p    0   0%  notrdy          empty
sg  150  --l------r    0  41%  ready    700088 idle
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) **t** Display Field Descriptions

Field	Description
ty	Device type.
eq	Equipment Number of the drive.
status	Device status. For a description of status codes, see Operator Display Status Codes.
act	Activity count.
use	Percentage of cartridge space used.
state	Current operating state of the removable media. Valid device states are as follows: • ready The device is on and the disk or tape is loaded in the transport; available for access. • notrdy The device is on but no disk or tape is present in the transport. • idle The device is not available for new connections. Operations in progress continue until completion. • off The device is not available for access. • down The device is available only for maintenance access. • nalloc The nalloc flag has been set, which prohibits any future allocation to this device. For more information, see Per-logical unit number (LUN) Allocation Control.
vsn	Volume serial name assigned to the volume, or the keyword nolabel if volume is not labeled. Blank if no volume is present in the transport, or device is off.

(T) - SCSI Sense Data Display

The T display shows the SCSI status of a SCSI device.
To invoke this display, type the following command:

```
Command:T
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **T** Display

Key	Function
Ctrl-b	Previous equipment
Ctrl-f	Next equipment

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

(u) - Staging Queue Display

The u display lists all files in the staging queue.
To invoke this display, type the following command:

```
Command:u
```

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **u** Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-k	Display the path on the second line of each entry
Ctrl-u	Half-page backward

Sample Display

This example shows the staging queue display.

Example: **samu**(1M) **u** Display

```
Staging queue by media type: all        samu 5.0     09:24:23 Sept 8 2008
     volumes 1 files 22

ty     length  fseq        ino   position     offset vsn

dt   451.611k    20       1030      207cc        473 DAT001
dt   341.676k    20       1031      207cc        7fc DAT001
dt   419.861k    20       1032      207cc        aa9 DAT001
dt   384.760k    20       1033      207cc        df2 DAT001
dt   263.475k    20       1034      207cc       10f5 DAT001
dt   452.901k    20       1035      207cc       1305 DAT001
dt   404.598k    20       1036      207cc       1690 DAT001
dt   292.454k    20       1037      207cc       19bb DAT001
dt   257.835k    20       1038      207cc       1c05 DAT001
dt   399.882k    20       1040      207cc       1e0b DAT001
dt   399.882k    40       1029      208d7          2 DAT001
dt   257.835k    40       1030      208d7        323 DAT001
dt   292.454k    40       1031      208d7        528 DAT001
dt   404.598k    40       1032      208d7        772 DAT001
dt   452.901k    40       1033      208d7        a9d DAT001
dt   263.475k    40       1034      208d7        e28 DAT001
dt   384.760k    40       1035      208d7       1038 DAT001
dt   419.861k    40       1036      208d7       133b DAT001
dt   341.676k    40       1037      208d7       1684 DAT001
dt   451.611k    40       1038      208d7       1931 DAT001
dt   161.326k    40       1039      208d7       1cba DAT001
dt   406.400k    40       1040      208d7       1dfe DAT001
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) **u** Display Field Descriptions

Field	Description
`ty`	Device type.
`length`	File length.
`fseq`	File system equipment number.

`ino`	The inode number.
`position`	The position of the archive file on the specific medium.
`offset`	Offset of the archive file on the specific medium.
`vsn`	Volume serial name of the volume.

(**U**) - Device Table Display

The U display shows the device table in a human-readable form.

- To display the device table for all devices, type the command with the following format:

```
Command:U
```

- To display the device table for a specific device, type the command with the following format:

```
Command:U <eq>
```

For eq, specify the Equipment Number of the device.

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **U** Display

Key	Function
Ctrl-b	Previous equipment
Ctrl-f	Next equipment

This display is designed for debugging. It is intended to be used only with the assistance of a Sun Microsystems support staff person.

Sample Display

This example shows the device table display.

Example:**samu**(1M) **U** Display

```
Device table: eq: 10    addr: 00000450  samu 5.0     09:28:40 Sept 8 2008

message:

0004000000014d58 0000000000000000       00000000 delay
0000000000000000 mutex                  00000000 unload_delay
00000aa8 next
73616d66 set:  samfs1
73310000
00000000
00000000
000a000a eq/fseq
08010801 type/equ_type
0000     state
00000000 st_rdev
00000000 ord/model
00000000 mode_sense
00000000 sense
00000000 space
00000000 capacity
00000000 active
00000000 open
00000000 sector_size
00000000 label_address
00000000 vsn:
00000000
00000000
00000000
00000000 status: ----------
00000000 dt
73616d66 name: samfs1
```

(v) - Automated Library Catalog Display

The v display shows the location and VSN of all disks or tapes currently cataloged in the automated library.

- To display the catalog for all devices, type the command with the following format:

```
Command:v
```

- To display catalog information for a specific device, type the command with the following format:

```
Command:v _eq_
```

For eq, specify the Equipment Number of the device. Type the keyword `historian` to view the historian catalog. For a list of all device names and Equipment Numbers, see (c) - Device Configuration Display.

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **v** Display

Key	Function
Ctrl -b	Page backward.
Ctrl -d	Next library catalog.
Ctrl -f	Page forward.

Ctrl -i	Detailed, two-line display format. When you enter Ctrl -i once, it shows times and barcodes. When you enter Ctrl -i a second time, it shows volume reservations on the second line.
Ctrl -k	Advance sort key. After you enter Ctrl-k, you can enter one of the following to select a sort key:
	1 - sort by slot.
	2 - sort by count.
	3 - sort by usage.
	4 - sort by VSN.
	5 - sort by access time.
	6 - sort by barcode.
	7 - sort by label time.
Ctrl-u	Previous automated library catalog.
/	Search for VSN.
%	Search for barcode.
$	Search for slot.

Sample Display

This example shows the automated library catalog display.

Example: **samu**(1M) **v** Display

```
Robot VSN catalog by slot       : eq 100samu     5.0     09:30:25 Sept 8 2008
    count 32
slot          access time count use flags         ty vsn

   0     2004/05/08 08:35   64   0% -il-o-b-----  sg 700071
   1     2004/05/08 09:08   27  12% -il-o-b-----  sg 700073
   2     2004/05/08 09:12   26  12% -il-o-b-----  sg 700077
   3     2004/05/08 08:39   37  40% -il-o-b-----  sg 700079
   4     2004/05/08 09:16   24   6% -il-o-b-----  sg 700084
   5     2004/05/08 09:18   24  41% -il-o-b-----  sg 700088
   6     none                0   0% -il-o-b-----  sg 700090
   7     none                0   0% -il-o-b-----  sg 700092
   8     none                0   0% -il-o-b-----  sg 000155
   9     none                0   0% -il-o-b-----  sg 000156
  10     none                0   0% -il-o-b-----  sg 000157
  11     none                0   0% -il-o-b-----  sg 000158
  12     none                0   0% -il-o-b-----  sg 000154
  13     none                0   0% -il-o-b-----  sg 000153
  14     none                0   0% -il-o-b-----  sg 000152
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) **v** Display Field Descriptions

Field	Description
`Robot VSN catalog`	Name of the specified automated library and time the display refreshed.
`count`	Number of slots allocated in this library's catalog.

slot	Slot number within the specified library.
access time	Time the volume was last accessed.
count	Number of accesses to this volume since the last audit operation.
use	Percentage of space used for the volume.
flags	Flags for the device. See the flag table for information about the flags.
ty	Device type.
vsn	Volume serial name of the volume.

Flags

The following table shows the flags using in the `flags` field. In some cases, more than one flag can occur in a field, and one flag overrides the other.

Flags Field for **samu**(1M) **v** Display

Flags	Description
A-----------	Volume needs audit.
}}{{{}I---------	Slot in use.
-L--------	Labeled. Overrides N.
-N--------	Unlabeled. This volume is foreign to the environment.
--E-------	Media error. Set when the software detects a write error on a cartridge.
--o-----	Slot occupied.
---C----	Volume is a cleaning tape. Overrides p.
---p----	Priority VSN.
----b---	Barcode detected.
-----W--	Write protect. Set when the physical write protection mechanism is enabled on a cartridge.
------R{-}--	Read only.
-------c{-}-	Recycle.
--------d{-}	Duplicate VSN. Overrides U.
--------U{-}	Volume unavailable.
-----------f	Archiver found volume full.
-----------X	Export slot.

(w) - Pending Stage Queue

The w display shows queued stage requests for which the volumes have not yet been loaded.

- To display the pending stage queue for all media, type the command with the following format:

```
Command:w
```

- To display the pending stage queue for a specific media type, type the command with the following format:

```
Command:w <mt>
```

For mt, specify one of the media types shown in the mcf(4) man page.

Navigation

The following table shows the control keys you can use in this display.

Control Keys for the **w** Display

Key	Function
Ctrl-b	Page backward
Ctrl-d	Half-page forward
Ctrl-f	Page forward
Ctrl-k	Display the path on the second line of each entry
Ctrl-u	Half-page backward

Sample Display

This example shows the pending stage queue.

Example:**samu**(1M) **w** Display

```
Pending stage queue by media type: all     samu     5.0     Thu Oct 11 13:20:27
    volumes 1 files 13

ty      length  fseq  ino  position  offset  vsn

at      1.383M     1   42     3a786    271b  000002
at      1.479M     1   56     3a786    5139  000002
at   1018.406k     1   60     3a786    6550  000002
at      1.000M     1   65     3a786    7475  000002
at      1.528M     1   80     3a786    99be  000002
at      1.763M     1   92     3a786    ce57  000002
at      1.749M     1  123     3a786   11ece  000002
at    556.559k     1  157     3a786   1532f  000002
at    658.970k     1  186     3a786   17705  000002
at    863.380k     1  251     3a786   1dd58  000002
at      1.268M     1  281     3a786   1f2b7  000002
at      1.797M     1  324     3a786   23dfa  000002
at      1.144M     1  401     3a786   2bb6d  000002
```

Field Descriptions

The following table shows the field descriptions for this display.

samu(1M) **w** Display Field Descriptions

Field	Description
`ty`	Device type.
`length`	File length.
`fseq`	File system Equipment Number.
`ino`	The inode number.

`position`	The position (in decimal format) of the archive file on the specific medium.
`offset`	Offset of the archive file on the specific medium.
`vsn`	Volume serial name of the volume.

Status Codes

The operator displays have different status codes for the removable media device displays and the file system displays. The following sections describe these status codes.

Removable Media Device Display Status Codes

The `o`, `r`, `s`, and `t` operator displays show status codes for removable media devices. Status codes are displayed in a 10-position format, reading from left (position 1) to right (position 10).
These status codes do not apply to the `samu`(1M) `f`, `m`, and `v` displays. For information about the status codes for the `f` and `m` displays, see File System Display Status Codes. For information about the status codes for the `v` display, see (v) - Automated Library Catalog Display.

The following table defines the status codes for each position.

Status Code for Removable Media Device Display

Status Bit	Meaning for a Device
`s---------`	Media is being scanned.
`m---------`	The automated library is operational.
`M---------`	Maintenance mode.
`}}{{{}E-------`	Device received an unrecoverable error in scanning.
`}}{{{}a-------`	Device is in audit mode.
`-l------`	Media has a label.
`-N------`	Foreign media.
`-L------`	Media is being labeled.
`--I-----`	Waiting for device to idle.
`--A-----`	Needs operator attention.
`--C---`	Needs cleaning.
`--U---`	Unload has been requested.
`---R--`	Device is reserved.
`----w{-}--`	A process is writing on the media.
`-----o{-}-`	Device is open.
`------P{-}`	Device is positioning (tape only).
`------F{-}`	For automated libraries, all storage slots occupied. For tape and magneto-optical drives, media is full.
`---------R`	Device is ready and the media is read-only.
`---------r`	Device is spun up and ready.
`---------p`	Device is present.
`---------W`	Device is write protected.

File System Display Status Codes

The f and m operator displays show status codes for file systems. Status codes are displayed in an 11-position format, reading from left (position 1) to right (position 11).
These status codes do not apply to the samu(1M) c, o, r, s, t, or v displays. For information about the status codes for the o, r, s, and t displays, see Status Codes for Removable Media Device Display . For information about the status codes for the v display, see (v) - Automated Library Catalog Display.

The following table defines the status codes for each position.

File System Display Status Codes

Status Bit	Meaning for a File System
m----------	File system is currently mounted.
M----------	File system is being mounted.
}}{{{}u--------	File system is being unmounted.
-A-------	File system data is being archived.
--R------	File system data is being released.
-=S=---	File system data is being staged.
--=1=--	Sun SAM file system version 1.
--=2=--	Sun SAM file system version 2.
---=c=-	Sun QFS shared file system.
----=W{-}--	Single writer.
-----=R{-}-	Multireader.
------=r{-}	mr devices.
----------d	md devices.

States of an Operator Display Device

The c, m, o, r, s, and t operator displays show state codes for each device. These codes represent the current access state for the device. The following table defines the state codes.

Operator Display Device States

Device State	Description
on	The device is available for access. For certain displays, this state might be superseded by the states ready or notrdy.
ro	The device is available for read-only access. For certain displays, this state might be superseded by the states ready or notrdy.
off	The device is not available for access. For tape and optical disk drives, possible reasons for the device to be in the off state include the following:\ • Cleaning was requested, but no cleaning cartridge was found in the automated library. • The cleaning cartridge cannot be loaded or unloaded from the drive. • Initialization found the drive status to be full, and attempts to clear the drive failed. • The system was unable to clear a cartridge from a drive. • Opening the drive for I/O failed during spin-up. • An error other than NOT READY was received when spinning down the drive for unloading. • Opening the standard tape driver on the drive failed during spin-up.
down	The device is available for maintenance access only.
idle	The device is not available for new connections. Operations in progress continue until completion.
ready	The device is on and the disk or tape loaded in the transport is available for access.

notrdy	The device is on, but no disk or tape is present in the transport.
unavail	The device is unavailable for access and cannot be used for automatic operations. You can continue to use the load(1M) and unload(1M) commands for moving media while the device is in the unavail state.
nalloc	The nalloc flag has been set, which prohibits any future allocation to this device. For more information, see Per-logical unit number (LUN) Allocation Control.

You can use the samu(1M) down, off, and on device state commands to change device states. You can enter these commands from any samu (1M) display, but if you enter them from the c, m, o, r, s, or t display, you can see the change in the display.
The following procedures show how to change a device's state from down to on and from on to down.

To Change a Drive State From **down** to **on**

1. Enter one of the following character keys to show a samu(1M) display with drive and automated library device states: c, m, o, r, s, or t.
2. Choose a device and verify that it is in the down state.
3. Use the :off command to stop all activity for the device.

```
Command:off <eq>
```

   where eq is the Equipment Number of the device.

4. Use the :on command.
   For example:

```
Command:on <eq>
```

   For eq, specify the Equipment Number of the device.

Operator Commands

The following topics describe the operator commands that you can enter from the samu(1M) operator utility's command interface. You can enter the commands from any display.

The following types of operator commands are available:

- Device Commands
- File System Commands: I/O Management
- File System Commands: Direct I/O Management
- File System Commands: Sun QFS Shared File Systems
- File System Commands: Miscellaneous
- Miscellaneous Commands
  If you want to enter any operator commands from the Solaris OS command line, you must use them as arguments to the samcmd(1M) command. For more information about the samcmd(1M) command, see the samcmd(1M) man page.

In the following subsections, each samu(1M) command is prefaced with a colon (:) character to indicate the entry is a a command and not a series of hot keys.

Device Commands

The following table shows the device commands and their actions.

Device Command Actions

Command	Action
down	Terminates operation on device eq.
idle	Restricts access to device eq by preventing new connections to the device. Existing operations continue until completion.

off	Logically turns off device eq.
on	Logically turns on device eq.
unavail	Selects device eq and makes it unavailable for use with the file system. You might set a drive state to `unavail`, for example, in a disaster recovery situation in which you are trying to load media to restore a file system and you do not want the Sun SAM software to attempt to use this drive.
unload	Unloads the mounted media for the specified removable media device eq. For magazine devices, the unload command unloads the mounted cartridge and ejects the magazine.
`nalloc`	Sets the `nalloc` flag on the device, which prohibits any allocation to this device. For more information, see Per-logical unit number (LUN) Allocation Control.
`alloc`	Removes the `nalloc` flag from the device. The `nalloc` flag prohibits any allocation to this device. The on command also removes this flag. For more information, see Per-logical unit number (LUN) Allocation Control.

All these commands use the following syntax:

```
:<command> <eq>
```

For eq, specify the Equipment Number of the device.

File System Commands: I/O Management

The following commands enable you to manage I/O characteristics dynamically.

The ***flush_behind*** Command

The `flush_behind` command sets the maximum `flush_behind` value. When set to a value greater than 0, modified pages that are being written sequentially are written to disk asynchronously to help the Solaris kernel layer keep the pages clean. By default, the maximum value is 0, which disables `flush_behind`.

```
:flush_behind <eq> <value>
```

For value, specify an integer number of kilobytes such that `0`≤ value≤ `8192`.
For eq, specify the Equipment Number for the file system.

The ***force_nfs_async*** and ***noforce_nfs_async*** Commands

These commands enable you to control whether the file system caches NFS data written to the server even if NFS has requested that the data be written synchronously through to disk. The `force_nfs_async` command caches NFS data.
The `force_nfs_async` command is effective only if the file system is mounted as an NFS server and only if the clients are mounted with the noac NFS mount option. For more information about mounting an NFS file system, see the `mount_nfs`(1M) man page.
The `noforce_nfs_async` command, which is the default, synchronously writes data through to disk.

```
:force_nfs_async <eq>
:noforce_nfs_async <eq>
```

For eq, specify the Equipment Number for the file system.

> Caution -
> The `force_nfs_async` option violates NFS protocols. Use this command with caution. In the event of a server interruption, data can be lost. Data is cached on the NFS server and cannot be seen immediately by all the clients if there are multiple NFS servers. Multiple NFS servers can be enabled within the Sun QFS shared file system. For more information about the Sun QFS shared file system, see Configuring a Sun QFS Shared File System.

The ***readahead*** Command

The `readahead` command specifies the maximum number of bytes that can be read ahead by the file system. The default contig number is 8 (131072 bytes).

```
:readahead <eq> <contig>
```

For eq, specify the Equipment Number for the file system.
For contig, specify units of 1-kilobyte blocks. This must be an integer such that 1 < contig < 8192. The contig specified is truncated to a multiple of 8 kilobytes.

For example, the following command sets the maximum contiguous block size to 262,144 bytes for the file system defined as Equipment Number 3:

```
:readahead 3 256
```

This value can also be configured in the `samfs.cmd` file by specifying the readahead directive. For more information, see the `samfs.cmd`(4) man page.

The ***sw_raid*** and ***nosw_raid*** Commands

These commands specify whether the file system aligns the writebehind buffer. Specify `sw_raid` if the software RAID feature of a package such as Solstice DiskSuite™ is also used on this file system. The default setting is `nosw_raid`.

```
:sw_raid <eq>
:nosw_raid <eq>
```

For eq, specify the Equipment Number for a file system.

The ***writebehind*** Command

The `writebehind` command specifies the maximum number of bytes that can be written behind by a file system. The default contig number is 8 (131072 bytes).

```
:writebehind <eq> <contig>
```

For eq, specify the Equipment Number for a file system.
For contig, specify units of 1-kilobyte blocks. This must be an integer such that 1 < contig < 8192.

For example, the following command sets the maximum contiguous block size to 262,144 bytes for the file system defined as Equipment Number `50`:

```
:writebehind 50 256
```

This value can also be configured in the `samfs.cmd` file by specifying the `writebehind` directive. For more information, see the `samfs.cmd` (4) man page.

The ***wr_throttle*** Command

The `wr_throttle` command sets the number of outstanding write kilobytes for one file. The default is the number of kilobytes that is 2% of the memory size.
If the percentage cannot be calculated, set the value to 100 MBytes. To determine memory size on Solaris:

```
npages = sysconf(_SC_PHYS_PAGES);

pagesizeb = sysconf(_SC_PAGESIZE);

fprintf(fp, "Memory size:\t\t%.1f MBytes (%.1f M2Bytes, %ld pages, %ld bytes/page)\n", ((float
)npages * (float)pagesizeb) / 1000000.0, ((float)npages * (float)pagesizeb) / 1048576.0, npages,
pagesizeb);
```

After you have found the memory size, set the value of wr_throttle to 2% of the total size, as shown in the following table:

Memory Size	Value for wr_throttle
1 GByte	20 MBytes
4 GBytes	80 MBytes
16 GBytes	320 MBytes
64 GBytes	1.3 GBytes

```
:wr_throttle <eq> <value>
```

For eq, specify the Equipment Number for a file system.
For value, specify an integer number of kilobytes. If value=0, there is no limit.

File System Commands: Direct I/O Management

The commands in this section control I/O on Sun QFS file systems. They enable you to change the type of I/O for an individual file based on I/O size and history. If direct I/O is specified for a file, for example, through the `setfa`(1) command, these options are ignored and all I/O to regular files is direct, if possible.

These commands refer to both well-aligned and misaligned I/O.

- Well-aligned I/O occurs when the file offset falls on a 512-byte boundary and when the length of the I/O transfer is at least 512 bytes.
- Misaligned_ I/O occurs when the file offset does not fall on a 512-byte boundary and the length of the transfer is less than 512 bytes.

For more information about I/O and I/O management, see [Advanced Topics|#ahdfv].

The ***dio_rd_form_min*** and ***dio_wr_form_min*** Commands

These commands set the lower limits for well-aligned I/O to the specified number of 1024-byte blocks. Use the `dio_rd_form_min` command to set the number for read operations, and use the `dio_wr_form_min` command to set the number for write operations. By default,the minimum number of blocks is 256.

```
:dio_rd_form_min <eq> <value>
:dio_wr_form_min <eq> <value>
```

For eq, specify the Equipment Number for the file system.
For value, specify an integer number of 1024-byte blocks to use for the lower limit. If value=0, automatic I/O switching is disabled.

The ***dio_rd_ill_min*** and ***dio_wr_ill_min*** Commands

These commands set the lower limit for misaligned I/O to the specified number of 1024-byte blocks. Use the `dio_rd_ill_min` command to set the number for read operations, and use the `dio_wr_ill_min` command to set the number for write operations. By default,the minimum number of blocks is 256.

```
:dio_rd_ill_min <eq> <value>
:dio_wr_ill_min <eq> <value>
```

For value, specify an integer number of 1024-byte blocks to use for the lower limit. If value=0, automatic I/O switching is disabled.

The ***dio_rd_consec*** and ***dio_wr_consec*** Commands

These commands set the number of consecutive I/O transfers that can occur with a buffer size greater than the specified lower limits. By default, value=0, which means that no default direct read operations occur based on I/O sizes.

```
:dio_rd_consec <eq> <value>
:dio_wr_consec <eq> <value>
```

For eq, specify the Equipment Number for the file system.
For value, specify the number of consecutive I/O transfers with a buffer size greater than the specified lower limit. The specified lower limit is the value of `dio_rd_form_min` for aligned read operations or `dio_rd_ill_min` for misaligned read operations.
For more information, see one or more of the following commands or mount parameters:

- The `dio_rd_form_min` and `dio_wr_form_min` Commands
- The `dio_rd_ill_min` and `dio_wr_ill_min` Commands

The ***dio_szero*** and ***nodio_szero*** Commands

These commands set or clear the direct I/O sparse zeroing mount option.
The `dio_szero` command causes uninitialized areas of sparse files written with direct I/O to be zeroed when the area is accessed. This makes the sparse file behavior the same as that for paged I/O. By default, sparse files written by direct I/O do not have the uninitialized areas zeroed for performance reasons. The default is `nodio_szero`.

```
dio_szero <eq>
nodio_szero <eq>
```

For eq, specify the Equipment Number for the file system.

The ***forcedirectio*** and ***noforcedirectio*** Commands

These commands enable you to control whether direct I/O is used as the default I/O mode. By default, the I/O mode is buffered and uses the page cache. The `forcedirectio` command enables direct I/O for all transfers. The `noforcedirectio` command restores the default,buffered I/O.
When direct I/O is specified, the system transfers data directly between the users buffer and disk. Use direct I/O only if the file system is used for large, block-aligned, sequential I/O.

```
:forcedirectio <eq>
:noforcedirectio <eq>
```

For eq, specify the Equipment Number for the file system.
For more information about I/O, see [Advanced Topics|#ahdfv].

File System Commands: Sun QFS Shared File Systems

The following file system commands are supported on Sun QFS shared file systems only.

The ***meta_timeo*** Command

The `metatimeo` command sets the time limit for the Sun QFS shared file system metadata cache. The default value is 3. For more information about using this feature, see Retaining Cached Attributes: the `meta_timeo`=n Option.

```
:meta_timeo <eq> <interval>
```

For eq, specify the Equipment Number of the file system.
For interval, specify time in seconds. After this interval expires, the client host systems obtain a new copy of the metadata information from the metadata server host.

The *mh_write* and *nomh_write* Commands

These commands enable or disable multihost read and write operations. For information about this feature, see Enabling Multiple Host Reads and Writes: the `mh_write` Option.

```
:mh_write <eq>
:nomh_write <eq>
```

For eq, specify the Equipment Number of the file system.

The *minallocsz* and *maxallocsz* Commands

These commands set the minimum and maximum block allocation size.

```
:minallocsz <eq> <value>
:maxallocsz <eq> <value>
```

For eq, specify the Equipment Number of the file system.
For value, and for more information about this feature, see Tuning Allocation Sizes: the `minallocsz=n` and `maxallocsz=n` Options.

The *rdlease*, *wrlease*, and *aplease* Commands

These commands control the amount of time granted for read, write, and append leases. The default time is 30 seconds. For information about this feature, see Using Leases in a Sun QFS Shared File System: the `rdlease=n`, `wrlease=n`, and `aplease=n` Options.

```
:rdlease <eq> <interval>
:wrlease <eq> <interval>
:aplease <eq> <interval>
```

For eq, specify the Equipment Number of the file system.
For interval, specify an integer number of seconds, `15`≤ interval≤ `600`. .

File System Commands: Miscellaneous

The following commands enable you to control leases, allocation sizes, and various other file system characteristics.

The *abr* and *noabr* Commands

These commands set or clear the application binary recovery (ABR) mount option.
For use in an Oracle RAC environment with Sun QFS asynchronous I/O (AIO) only. These mount options disable or enable ABR of software mirrors. They apply only to Sun QFS file systems built on Solaris Volume Manager mirrored volumes that support ABR.

```
:abr <eq>
:noabr <eq>
```

For eq, specify the Equipment Number for the file system.

The *dmr* and *nodmr* Commands

These commands set or clear the direct mirror reads (DMR) mount option.
For use in an Oracle Real Application Cluster (RAC) environment with Sun QFS AIO only. These mount options disable or enable DMR of software mirrors. They apply only to Sun QFS file systems built on Solaris Volume Manager mirrored volumes that support DMR.

```
:dmr <eq>
:nodmr <eq>
```

For eq, specify the Equipment Number for the file system.

The *invalid* interval Command

The `invalid` command specifies that the file system hold cached attributes for at least the specified number of seconds after a file is modified. You can specify this command only if the file system was mounted originally with the `reader` mount option. For information about mount options, see the `mount_samfs`(1M) man page.

```
:invalid <eq> <interval>
```

For eq, specify the Equipment Number for the file system.
For interval, specify the number of seconds to hold the attributes after file modification. For example, assume that interval=`30`. When you issue an `ls`(1) command, you might not see a newly created file appear in its output for 30 seconds after it has been created on its writer host.

The *mm_stripe* Command

The `mm_stripe` command sets the metadata stripe width for the file system to the specified number of 16-kilobyte disk allocation units (DAUs). The default is 1 DAU so that the file system writes one DAU of metadata to one LUN before switching to another LUN.

```
:mm_stripe <eq> <value>
```

For eq, specify the Equipment Number of the file system.
For value, specify either `0` or `1`. If value=`1`, which is the default, the file system writes one DAU of metadata to one LUN before switching to another LUN. If value=`0`, the metadata is round-robined across all available metadata LUNs.

The *qwrite* and *noqwrite* Commands

The `qwrite` and `noqwrite` commands control the ability to perform simultaneous read and write operations to the same file from different threads. Specify `qwrite` only if file system users handle multiple simultaneous transactions to the same file. This is useful in database applications. The `qwrite` feature improves I/O performance by queuing multiple requests at the drive level. The `qwrite` specification is disabled for NFS reads or writes of the file system.

The default setting is `noqwrite`, so the file system disables simultaneous read and write operations to the same file. This is the mode defined by the UNIX `vnode` interface standard that gives exclusive access to only one writer and forces other writers and readers to wait.

```
:qwrite <eq>
:noqwrite <eq>
```

For eq, specify the Equipment Number of the file system.

The *refresh_at_eof* and *norefresh_at_eof* Commands

The `refresh_at_eof` and `norefresh_at_eof` commands can be used for fast updates to Sun QFS hosts that are mounted with the `reader` mount option in a multireader file system. This option ensures that the system refreshes the current file size when the read buffer exceeds the end of file. You can use this, for example, if the writer host system is appending to a file and the reader is issuing `tail`(1) commands with the `-f` option. The default is `norefresh_at_eof`.

```
:refresh_at_eof <eq>
:norefresh_at_eof <eq>
```

For eq, specify the Equipment Number of the file system.

The *suid* and *nosuid* Commands

The `suid` and `nosuid` commands control whether running programs are allowed to automatically change their owner IDs. For more information about the implications of using these mount options, see the `suid` and `nosuid` mount option descriptions on the `mount_ufs`(1M) man page and see the `suid`(2) man page.

```
:suid <eq>
:nosuid <eq>
```

For eq, specify the Equipment Number of the file system.

The *stripe* Command

The stripe command sets the stripe width for the file system to the specified number of disk allocation units (DAUs). The stripe width specifies that value multiplied by the DAU bytes are written to one LUN before switching to the next LUN. You can use the sammkfs(1M) -a command to set the DAU size on the file system when it is initialized.

```
:stripe <eq> <value>
```

For eq, specify the Equipment Number of the file system.
For value, specify an integer such that 0 < value < 255. If value=0, files are round-robin on each slice. The default value on file systems with an ms Equipment Type and on file systems with an ma Equipment Type with no striped group (gXXX) components is as follows:

- 128 kilobytes/DAU for DAUs < 128 kilobytes
- 1 for DAUs > 128 kilobytes

By default, value=0 on a Sun QFS shared file system.

By default, value=0 on file systems with an ma Equipment Type with any striped group (gXXX) components.

The system sets value=0 if mismatched striped groups exist.

For more information about file system types, see Design Basics and Configuring the File System.

The *sync_meta* Command

The sync_meta command determines whether metadata is written to disk every time it changes. If you are using this command on Sun QFS shared file system, also see Specifying the Frequency With Which Metadata Is Written: the sync_meta=n Option.

```
:sync_meta <eq> <value>
```

For eq, specify the Equipment Number of the file system.
For value, specify either 0 or 1, as follows:

- For value is 0, metadata is held in a buffer after it changes. For an unshared Sun QFS file system in which higher performance is desired, you can set value to 0. In this case, the system performs a delayed write operation in which metadata is held in a buffer before it is written to disk. This is the default for unshared file systems and for file systems that are not mounted as multireader file systems.
- For value is 1, metadata is written to disk every time it changes. This slows performance, but it increases data consistency. This is the default for Sun QFS file systems mounted as multireader file systems or as shared file systems. For a Sun QFS shared file system, value must be set to 1 if failover capability is required.

The *trace* and *notrace* Commands

The trace command enables tracing for a file system. The notrace command disables tracing. These are global directives that affect all operations. For more information about file system tracing, see the defaults.conf(4) man page.

```
:trace <eq>
:notrace <eq>
```

For eq, specify the Equipment Number of a file system.

Miscellaneous Commands

The following commands enable you to control tracing, open access to a disk device, and perform several other miscellaneous tasks.

The *clear* vsn Command

The clear command clears the specified VSN from the Removable Media Mount Requests display. For more information, see (p) - Removable Media Load Requests Display.

```
:clear <vsn>
:clear <vsn> <index>
```

For vsn, specify the volume to mount. Any process waiting for the VSN mount is aborted.
For index, specify the decimal ordinal of the VSN in the removable media display.

The *devlog* Command

The devlog command sets one or more events to be logged.

```
:devlog <eq>
:devlog <eq> <option>
```

For eq, specify the Equipment Number of a device.
For option, specify one or more event types. Possible event types are as follows: all, date, default, detail, err, event, label, mig, module, msg, none, retry, stage, syserr, and time. For information about these options, see the defaults.conf(4) man page.
If no option is specified, the system does not change the current events being logged for the eq specified.

The *diskvols* flag Command

The diskvols command sets or clears flags in the disk volume dictionary.

```
:diskvols <volume> +flag
:diskvols <volume> -flag
```

For volume, specify the volume in the disk volume dictionary.
For flag, specify one of the five flags in the D samu(1M) display. For information about the disk volume dictionary and the flags see (D) - Disk Volume Dictionary or the samu(1M) man page.

The *dtrace* Commands

The dtrace commands control the dtrace feature for one or more processes. The dtrace commands specify various tracing options.

```
:dtrace <daemon_name> on
:dtrace <daemon_name> off
:dtrace <daemon_name>.<variable> <value>
```

For daemon_name, specify the keyword all to affect all processes or a process name. If one of the following process names is specified, the tracing command affects that process only: sam-archiverd,sam-catserverd, sam-fsd, sam-rftd, sam-recycler, sam-sharefsd, and sam-stagerd.
For variable and value, specify one of the following variable and value pairs. The defaults.conf(4) man page contains comprehensive information about these arguments.

- file value
  Specify the name of a file to which trace files can be written. This can be a full path name.
- options value
  Specify a space-separated list of trace options.
- age value
  Specify the trace file rotation age. Note: Do not set an age of two minutes or less. If you do, the rotation never happens.
- size value
  Specify the size of the trace file at which rotation begins.

The *fs* Command

The `fs` command sets the file system to be displayed through the `N` display.

```
:fs <fsname>
```

For fsname, specify the name of the file system to be examined.

The *mount* Command

The `mount` command selects a Sun QFS file system.

```
:mount <mntpt>
```

For mntpt, specify the mount point of a file system.

The *open* Command

The `open` command enables access to the specified disk device. You must issue this command before you can use the `read` command, disk sector display (S), or file label display (F).

```
:open <eq>
```

For eq, specify the Equipment Number of a device.

The *read* Command

The `read` command reads the specified sector from the currently opened disk device. You must open the device before it can be read.

```
:read <addr>
```

For addr, specify the hexadecimal sector address.

The *refresh* Command

The `refresh` command sets the amount of time between `samu`(1M) screen refreshes.

```
:refresh <i>
```

For i, specify a time in seconds.

The *snap* Command

The `snap` command sends a snapshot of a display window to file. The default file is `snapshots` in the current working directory. To aid in problem reporting, take a snapshot of all the `samu`(1M) utility's displays. Each new snapshot is appended to the snapshots file. The file can be printed, examined using `vi`(1), or faxed to Sun Microsystems customer support staff.

```
:snap
:snap <filename>
```

For filename, specify the path of a file to receive the display information.

The *!* shell_command Command

The `!` command enables you to run a shell command without leaving the `samu`(1M) operator utility.

```
:! <shell_command>
```

For shell_command, specify a command.

Sun Storage Archive Manager (SAM-QFS) Configuration and Administration Guide

Sun Storage Archive Manager (SAM-QFS) Configuration and Administration Guide

Contents

More About SAM-QFS

The Sun Storage Archive Manager (SAM-QFS) software provides a configurable file system with storage, archive management, and retrieval capabilities. The SAM-QFS software archives files by copying the files from online disk cache to archive media. The archive media can consist of disk slices in another file system or it can consist of removable tape or magneto-optical cartridges in automated or manually loaded storage devices. The SAM-QFS software automatically maintains online disk space at site-specified usage thresholds. It releases disk space associated with archived file data and restores the files to online disk when they are needed.

Components of SAM-QFS

The SAM-QFS software consists of the following components which define the archiving lifecycle:

- Archiving
- Releasing
- Staging
- Recycling

Archiving

The archiver automatically copies online disk cache files to archive media. The archive media can consist of either online disk files or removable media cartridges. The archiver requires you to configure the `archiver.cmd` file to define what to archive. You can configure the archiver to create up to four archive copies on a variety of archive media. If a file is segmented, each segment is treated as a file and is archived separately. The archiving process is initiated after disk-based files match a site-definable set of selection criteria.

For more information about the archiver, see About Archiving.

Releasing

The releaser automatically maintains the file system's online disk cache at site-specified percentage usage thresholds by freeing disk blocks occupied by eligible archived files.

Releasing is the process of freeing primary (disk) storage that is used by an archived file's data. Two threshold values, the high-water mark and the low-water mark, both expressed as a percentage of total disk space, are used to manage free space in the online disk cache. When online disk consumption exceeds the high-water mark, the system automatically begins releasing the disk space occupied by eligible archived files. Disk space occupied by archived file data is released until the low-water mark is reached.

Files are selected for release depending on their size and age. The first portion of a file can be retained on disk for speedy access and for masking staging delays. If a file has been archived in segments, portions of the file can be released individually.

For more information about the releaser, see About Releasing.

Staging

The stager restores file data to the disk cache. When a user or process requests file data that has been released from disk cache, the stager automatically copies the file data back to the online disk cache.

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

The SAM-QFS software processes stage request errors automatically. If a stage error is returned, the system attempts to find the next available archive copy of the file. Stage errors that can be automatically processed include media errors, unavailability of media, unavailability of an automated library, and others.

For more information about staging, see About Staging.

Recycling

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

As users modify files, archive copies associated with the old versions of these files are considered to be expired on their archive media. Because these copies are no longer needed, they can be purged from the system. The recycler identifies the archive volumes with the largest proportions of expired archive copies and preserves the unexpired copies by moving them to separate volumes. The recycling process is transparent to end users.

If a removable media volume contains only expired copies, you can take one of the following actions:

- Relabel the volume for immediate reuse.
- Export the volume to offsite storage as a historical record of file changes. You can use standard UNIX utilities to restore previous versions of files from expired archive copies.

For more information about recycling, see About Recycling.

Supported Storage Devices

The SAM-QFS environment supports a wide variety of tape storage and magneto-optical devices. The automated libraries that SAM-QFS supports can be divided into the following groups, depending on how they are attached to the environment:

- A direct attachment. A direct attached library is connected directly to the host system using a Small Computer System Interface (SCSI). This connection can be either a direct connection or a Fibre Channel connection. For example, a direct attachment is used for Sun StorageTek libraries. The SAM-QFS system controls these libraries directly using the SCSI standard for automated libraries.
- A network attachment. The SAM-QFS software can be configured as a client of the library's host system. The network attached libraries include some of the StorageTek, ADIC/Grau, IBM, and Sony libraries. These libraries use a software package supplied by the vendor. In these cases, the SAM-QFS software interfaces with the vendor software, using a daemo designed for the automated library.

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

When possible, the system uses the standard Solaris disk and tape device drivers. For devices not directly supported in the Solaris OS, such as certain library and optical disk devices, the SAM-QFS software packages include special device drivers.

For a list of supported storage devices, contact your Oracle sales representative or your authorized service provider (ASP). For information about configuring storage devices, see Configuring Storage Devices for Archiving.

SAM-Remote Software

Sun SAM-Remote software is a client/server implementation that enables libraries and other removable media devices to be shared between SAM-QFS host systems. Sun SAM-Remote software enables you to configure multiple storage clients that archive and stage files from a centralized tape library or magneto-optical library. For example, if you have host systems on a network that spans a large geographical area, files created in one city can be archived to cartridges in a library located miles away.

For more information see Using the Sun SAM-Remote Software.

Configuring Storage Devices for Archiving

Perform the tasks in this section if you plan to enable archiving to tape or magneto-optical media. If you plan to archive to disk, you do not need to perform these tasks.

This section describes how to verify and update the following files:

- `/kernel/drv/st.conf`, which configures tape drives attached to the server through a SCSI or FC attachment.
- `/kernel/drv/samst.conf`, which configures the following devices that the Sun Storage Archive Manager (SAM) software recognizes by default:
    - Direct attached automated libraries
    - Magneto-optical drives attached to the server through a SCSI or FC attachment

The SAM package includes the `/opt/SUNWsamfs/examples/st.conf_changes` file, which contains configuration information for tape drives that are not supported in the Solaris kernel by default.

Task Map: Configuring Storage Devices for Archiving

Depending on the devices you want to configure, you must complete one or more of the following procedures.

Step	Task	Description
1	Create List of Devices	Take an inventory of your devices to configure.
2	Add Tape Devices	Perform this task for each tape drive that you want to add to the SAM environment.
3	Add Tape Drive Interface Target IDs and LUNs	Perform this task if your tape drives are attached through a SCSI or FC interface.

4	Add Libraries or Magneto-Optical Drives	Perform this task if you have any magneto-optical drives, SCSI-attached automated libraries, or FC-attached automated libraries that you want to include in your SAM environment.
5	Verify Configured Devices	Verify that you have all of your devices configured properly.
6	Enable the Storage Device Configuration	Reboot the system to enable the changes you have made.
7	Create Parameters Files for Network Attached Automated Libraries	Create parameter files for the network attached automated libraries.

The procedures in this task include an example that is based on the inventory list shown in the following table.

Example Inventory List - Devices to Be Configured

Device Name, Manufacturer, and Model	Target ID	LUN	Node WWN
SCSI-attached tape drives			
QUANTUM DLT7000	1	0	Not applicable
QUANTUM DLT7000	2	0	Not applicable
FC-attached tape drives			
StorageTek 9840	Not applicable	0	500104f00043abfc
StorageTek 9840	Not applicable	0	500104f00045eeaf
IBM ULT3580-TD1	Not applicable	0	500104f000416304
IBM ULT3580-TD1	Not applicable	0	500104f000416303
SCSI-attached automated libraries			
StorageTek 9730	0	0	Not applicable
FC-attached automated libraries			
StorageTek L700	Not applicable	0	500104f00041182b

Note -
The device names are shown as they appear in the discovery output.

How To Add Tape Devices for Archiving

Perform this procedure to include your tape drives in your archiving environment. In this procedure, you make an entry in the `st.conf` file for each unique tape drive that is on your inventory list.

Tip -
You can also add tape devices from SAM-QFS Manager. For information, see How to Add Tape Devices From SAM-QFS Manager.

1. Copy `/kernel/drv/st.conf` to a backup file. For example:

```
# cp /kernel/drv/st.conf /kernel/drv/st.conf.orig
```

2. Open the `/kernel/drv/st.conf` file and find the following string:

```
#tape-config-list=
```

3. Remove the pound character (#).
4. Open the `/opt/SUNWsamfs/examples/st.conf_changes` file. For each tape drive on your inventory list, do the following.
   a. Search for the device definition entry.
      For example, searching for the Quantum DLT 7000 tape drive that is in the example inventory locates the following entry:

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

   b. Copy the entry from the `st.conf_changes` file to the `st.conf` file, placing it after the `tape-config-list` line. The following example shows the resulting `st.conf` file.

      ```
      ...
      tape-config-list=
      "QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape",
      ...
      ```

   c. Note the final string in the entry, which is enclosed in quotation marks. In this example, the final string is "`dlt7-tape`". This string is the tape configuration value.
   d. Search `/opt/SUNWsamfs/examples/st.conf_changes` to find the line that begins with this tape configuration value. In this example, the value is:

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

   e. Copy the tape configuration value to the `st.conf` file, placing it after the device definition line. The following example shows the lines now contained in the `st.conf` file.

      ```
      ...
      tape-config-list=
      "QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape";
      dlt7-tape =    1,0x36,0,0xd679,4,0x82,0x83,0x84,0x85,3;
      ...
      ```

   f. Repeat this procedure for each type of tape drive.

      Note -
      In the `st.conf_changes` file, a tape configuration value is repeated for each device definition that uses the same tape configuration. In the `st.conf` file, include only one entry for each tape configuration value. For example, the Sony SDT-5000 and the Sony SDT-5200 tape drives both use "DAT" as the final string. A single entry for the DAT tape configuration value is sufficient.

5. Replace the comma (`,`) at the end of the last device definition line with a semicolon (`;`).
   The following example shows a `st.conf` file with definitions for the Quantum DLT 7000, the StorageTek 9840, and the IBM ULT3580 tape drives. The semicolon is after "`CLASS_3580`"

   ```
   ...
   tape-config-list=
   "QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape",
   "STK     9840", "STK 9840 Fast Access", "CLASS_9840",
   "IBM     ULT3580-TD1",  "IBM 3580 Ultrium",     "CLASS_3580";
   dlt7-tape =    1,0x36,0,0xd679,4,0x82,0x83,0x84,0x85,3;
   CLASS_9840 = 1,0x36,0,0x1d679,1,0x00,0;
   CLASS_3580 = 1,0x24,0,0x418679,2,0x00,0x01,0;
   ...
   ```

6. Save your changes.
7. Close the `/opt/SUNWsamfs/examples/st.conf_changes` file.
   Do not close the `st.conf` file because you continue to edit the file in the next procedure.
8. Go to the #How To Add Tape Drive Interface Target IDs and LUNs for Archiving procedure.

How To Add Tape Drive Interface Target IDs and LUNs for Archiving

For each tape drive on your hardware inventory list that is attached through a SCSI or FC interface, you must confirm that an entry in the `st.conf` file defines that interface. This procedure shows how to verify and, if necessary, add target ID and LUN entries.

> Note -
> Do not use this procedure to add interface information for magneto-optical drives. See
> How To Add Libraries or Magneto-Optical Drives for Archiving

1. Open the file `/kernel/drv/st.conf` if it is not already open.
2. To use the SCSI interface to attach tape drives, do the following:
   a. Search for entries that have the following format to locate the list of SCSI target IDs and LUNs:

      ```
      name="st" class="scsi" target=<target> lun=<lun>;
      ```

      target is the target ID for each SCSI drive found.
      lun is the corresponding LUN for each SCSI drive found.

   b. Within the list, find each entry that corresponds to each SCSI target and LUN on your hardware inventory list. The following example shows the two entries that correspond to the two Quantum DLT 7000 drives that are attached to LUN 0 and have target IDs 1 and 2, shown in Example Inventory List.

      ```
      name="st" class="scsi" target=1 lun=0;
      name="st" class="scsi" target=2 lun=0;
      ```

      Note that an entry might extend over two lines.

      - If the entry is preceded by a pound character (#), delete the character. A pound character marks a line as a comment.
      - If the entry is missing, create an entry for the SCSI target and LUN line you need, following the format shown in Step a and using the information in your hardware inventory list.
3. To use the FC interface to add tape drives and if you are not using the Sun StorageTek SAN Foundation Software I/O stack, create a line for each FC-attached device after the SCSI target ID and LUN list, using the following format:

   ```
   name="st" parent="fp" lun=<lun> fc-port-wwn="<world-wide-name>"
   ```

   For lun, specify the LUN for the drive.
   For world-wide-name, specify the World Wide Name (WWN) for the drive.
   The following example shows the lines that support the StorageTek 9840 and IBM ULT3580 tape drives included in the Example Inventory List.

   ```
   name="st" parent="fp" lun=0 fc-port-wwn="500104f00043abfc"
   name="st" parent="fp" lun=0 fc-port-wwn="500104f00045eeaf"
   name="st" parent="fp" lun=0 fc-port-wwn="500104f000416304"
   name="st" parent="fp" lun=0 fc-port-wwn="500104f000416303"
   ```

4. Save your changes and close the `st.conf` file.
5. To include magneto-optical drives or automated libraries attached through a SCSI or a Fibre Channel interface, go to the How To Add Libraries or Magneto-Optical Drives for Archiving procedure.

How to Add Tape Devices From SAM-QFS Manager

To add a tape device from the SAM-QFS Manager:

1. In the left pane, click Archive Media.
2. In the Tape Library Summary window, click Add.
3. Follow the steps in the wizard.

How To Add Libraries or Magneto-Optical Drives for Archiving

The `inquiry.conf` file lists all devices that are supported.
The `/kernel/drv/samst.conf` file contains a list of SCSI and FC entries and works with the `/opt/SUNWsamfs/examples/inquiry.conf` file to define the devices that can be included in a SAM environment.
The following procedures show how to verify the entries in the `samst.conf` and to update the file if necessary.

You update the `samst.conf` file, depending on your environment:

- If you have only network-attached automated libraries, you do not need to verify device support or update it.
- If you use a SCSI or FC interface to attach a tape library to the server, use the procedure in How To Configure Device Support in SCSI or FC Environments to discover the tape libraries with the current drive information.
- If you have a direct attached library with a target number larger than 6 or a LUN identifier larger than 1, use the procedure in How To Configure Device Support for a Direct Attached Library.

How To Configure Device Support in SCSI or FC Environments

Repeat this procedure for each device in your environment.

1. On the SAM-QFS Manager Managed Hosts page, select the name of the server to which you want to add a library.
   The File Systems Summary page is displayed.
2. Expand the Archive Media section and select Tape Libraries.
   The Tape Library Summary page is displayed.
3. Click Add to start the Add a Library wizard.
4. Follow the steps to add the device.
   When you have completed the wizard steps, the `samst.conf` file is automatically updated.

How To Configure Device Support for a Direct Attached Library

1. Copy the `/kernel/drv/samst.conf` file to a backup file. For example:

```
# cp /kernel/drv/samst.conf /kernel/drv/samst.conf.orig
```

2. Edit the `/kernel/drv/samst.conf` file.
3. To include SCSI-attached magneto-optical drives or SCSI-attached libraries, do the following:
   a. Search for entries that have the following format to locate the list of SCSI targets and LUNs:

   ```
   name="samst" class="scsi" target=<target> lun=<lun>;
   ```

   target is the target ID for each SCSI drive found.
   lun is the corresponding LUN for each SCSI drive found.

   b. Within the list, find the entry that corresponds to each SCSI target ID and LUN on your inventory list.
   For example, the StorageTek 9730 automated library is attached to target 0 and LUN 0. The following line corresponds to that interface:

   ```
   name="samst" class="scsi" target=0 lun=0;
   ```

   Note that an entry might extend over two lines if it contains return characters.

   - If the entry starts with a pound character (#), delete the character. A pound (#) character marks a line as a comment.
   - If the entry is missing, create an entry for the SCSI target and LUN line you need, following the format shown in Step a and using the information in your hardware inventory list.
4. To include FC-attached magneto-optical drives or FC-attached automated libraries, create a line for each FC-attached one in your inventory list. Place the lines at the end of the SCSI target and LUN list, using the following format:

```
name="samst" parent="fp" lun=<lun> fc-port-wwn="<world-wide-name>"
```

For lun, specify the LUN for the drive.

For world-wide-name, specify the WWN for the drive.
The following example shows the line added to support the StorageTek L700 tape drive in the Example Inventory List.

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

5. Save your changes and exit the `samst.conf` file.

Verifying That All Devices Are Configured

1. Use the `cfgadm`(1M) command to list the devices included in the SAM environment. For example:

```
# cfgadm -al
Ap_Id                           Type         Receptacle   Occupant     Condition
c0                              scsi-bus     connected    configured   unknown
c0::dsk/c0t6d0                  CD-ROM       connected    configured   unknown
c1                              fc-private   connected    configured   unknown
c1::500000e0103c3a91            disk         connected    configured   unknown
c2                              scsi-bus     connected    unconfigured unknown
c3                              scsi-bus     connected    unconfigured unknown
c4                              scsi-bus     connected    configured   unknown
c4::dsk/c4t1d0                  disk         connected    configured   unknown
c4::dsk/c4t2d0                  disk         connected    configured   unknown
c5                              fc-fabric    connected    configured   unknown
c5::100000e00222ba0b            disk         connected    unconfigured unknown
c5::210000e08b0462e6            unknown      connected    unconfigured unknown
c5::210100e08b2466e6            unknown      connected    unconfigured unknown
c5::210100e08b27234f            unknown      connected    unconfigured unknown
c5::500104f00043abfc            tape         connected    configured   unknown
c5::500104f00043bc94            tape         connected    configured   unknown
c5::500104f00045eeaf            tape         connected    configured   unknown
c5::500104f000466943            tape         connected    configured   unknown
c5::500104f00046b3d4            tape         connected    configured   unknown
c5::500104f0004738eb            tape         connected    configured   unknown
c6                              fc           connected    unconfigured unknown
c7                              scsi-bus     connected    unconfigured unknown
c8                              scsi-bus     connected    unconfigured unknown
usb0/1                          usb-kbd      connected    configured   ok
usb0/2                          usb-mouse    connected    configured   ok
usb0/3                          unknown      empty        unconfigured ok
usb0/4                          unknown      empty        unconfigured ok
```

2. Examine the output to make sure that it shows all the devices you want configured in your SAM environment.
If a device is not configured, use the `cfgadm`(1M) command to configure it. For more information, see the `cfgadm`(1M) man page.
Because of a bug in the `cfgadm`(1) command, you might receive an error similar to the following:

```
# cfgadm -c configure -o force_update c4::500104f000489fe3
cfgadm: Library error: failed to create device node: 500104f00043abfc: Device busy
```

Despite the error, the `cfgadm`(1M) command processes the request.

Enable Storage Device Configuration

You must reboot the system to have the changes you have made to the `st.conf` and `samst.conf` files take effect.

```
# reboot -- -r
```

Handling Errors in the **st.conf** File

Errors can occur if the `st.conf` file is not configured properly during SAM software installation.

The following messages in the sam-log file indicate that the appropriate changes have not been made to /kernel/drv/st.conf. Follow the steps in How To Add Tape Devices to the /kernel/drv/st.conf File to fix the error.

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

The following device log messages correspond to the sam-log message:

```
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
```

Creating Parameters Files for Network Attached Automated Libraries

Perform the applicable procedure in this section if you have network attached automated libraries in your Sun Storage Archive Manager (SAM) environment. You must configure your storage devices first, as described in Configuring Storage Devices for Archiving.

You can include automated libraries in a SAM environment either by directly attaching them to the server or by attaching them to the environment's network. Libraries attached through a SCSI or Fibre Channel (FC) attachment are direct attached libraries. Libraries attached through a network attachment are network attached libraries. In this task, you create a parameters file for each network attached library to be included in your environment.

> Note -
> The examples and the discussions in the following sections mention both the parameters files for network attached automated libraries and the mcf file. The mcf file is the main configuration file for the SAM software. You created your mcf file in Configuring the File System Environment. The parameters file and the mcf file are both mentioned in this section because the two files reference each other.

How To Configure an ADIC/Grau Automated Library Parameters File

The ADIC/Grau automated library operates within SAM environments through the grauaci interface. This interface uses the DAS/ACI 3.12 interface supplied by ADIC/Grau. For more information about the DAS/ACI interface, see your ADIC/Grau documentation.

> Note -
> ADIC/Grau network attached libraries are not supported by the SAM software on an x64 hardware platform.

Before You Begin

Ensure that the following are true and the ADIC/Grau automated library is prepared for inclusion in a SAM environment.

- The ADIC/Grau automated library is operational.
- The ADIC/Grau library is operating on the DAS (Distributed AML Server).
- Both the avc (avoid volume contention) and the dismount parameters are set to true in the DAS configuration file for this client.

Follow this procedure to create a parameters file for each ADIC/Grau library that you want to configure.

1. Change to the /etc/opt/SUNWsamfs directory.
   The parameters file can be written to any directory, but the most common location is /etc/opt/SUNWsamfs.

> Note -
> When you created your `mcf` file, in Configuring the File System Environment, you included the full path name to the parameters file in the `mcf` file. Make sure that the `mcf` file points to the correct location for the parameters file that you create in this procedure.

2. Open a new file abd give it a name that corresponds to the library that you are configuring. For example:

```
# vi grau50
```

3. Type a list of keyword = value parameter lines in the ADIC/Grau parameters file.
   The various keyword values identify the ADIC/Grau automated libraries, the drives associated with the libraries, and the server name. All keyword and value entries are case-sensitive, so enter them exactly as specified in the DAS configuration file and in the SAM `mcf` file.
   The following table shows the keyword = value parameters that must appear in the ADIC/Grau parameters file.

Parameter	Meaning
client = client-id	The name of the client as defined in the DAS configuration file. This parameter is required.
server = server-id	The host name of the server running the DAS server code. This parameter is required.
acidrive drive-id = path	The name of the drive as configured in the DAS configuration file. For path, specify the path to the drive as entered in the Equipment Identifier field of the SAM `mcf` file. Include one `acidrive` line for every drive assigned to the client.

   Comments can appear anywhere on any line, and they must begin with a pound sign (#). The system ignores characters to the right of the pound sign.

If the ADIC/Grau library contains various media types, a media changer exists for each media type. Each media changer has a unique client name in the DAS configuration, a unique library catalog, and a unique parameters file.

The sample ADIC/Grau parameters file shown in the following example defines one ADIC/Grau automated library supporting DLT tape and one ADIC/Grau automated library supporting a Hewlett-Packard optical drive.

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

> Note -
> Configuration Example for a Shared File System on a Solaris OS Platform shows the `mcf` file that corresponds with the example of a ADIC/Grau network attached automated library parameters file that was created in this procedure. The example `mcf` file points to the `grau50` file in the `/etc/opt/SUNWsamfs` directory.

The following directory contains diagnostic information that can be useful for troubleshooting:

`/var/opt/SUNWsamfs/.grau`

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

How To Configure an IBM 3494 Automated Library Parameters File

The IBM 3494 automated tape library operates in SAM environments with the assistance of the IBM `lmcpd` daemon package. You can obtain the IBM `lmcpd` daemon package from IBM.

> Note -
> IBM 3494 network attached libraries are not supported by SAM software on an x64 hardware platform.

Before You Begin

Ensure that the following are true and the IBM 3494 automated library is prepared for inclusion in a SAM environment.

- The IBM 3494 automated library is operational.
- The IBM `lmcpd` daemon package is installed and working.
- The `/etc/ibmatl.conf` file is configured and working.
- The IBM 3494 automated library can be used as a single physical library or as multiple logical libraries. If you divide this library into multiple logical libraries, create a parameters file for each logical library.

Follow this procedure to create a parameters file for each physical or logical library that you want to include in the SAM environment.

1. Change to the `/etc/opt/SUNWsamfs` directory. The parameters file can be written to any directory, but the most common location is `/etc/opt/SUNWsamfs`.

   > Note -
   > When you created your `mcf` file in Configuring the File System Environment, you included the full path name to the parameters file in the `mcf` file. Make sure that the `mcf` file points to the correct location for the parameters file that you create in this procedure.

2. Create a new file and give it a name that corresponds to the library that you are configuring. For example:

```
# vi ibm50
```

3. Type a list of keyword = value and pathname = value pairs in the IBM 3494 parameters file.

All arguments are case-sensitive. The following table shows how to specify the parameters.

Parameter	Meaning
`name` = name	The name assigned by you as system administrator, and specified in the `/etc/ibmatl.conf` file. This name is also the symbolic name of the library. This parameter must be supplied. There is no default.
`category` = hexnumber	A hexadecimal number between `0x0001` and `0xfeff`. By default, the SAM software sets this value to 4 for media under its control. If you have divided your physical library into multiple logical libraries, make sure that the value of hexnumber is different in each logical library. This parameter determines which tapes are assigned to which library. When you import media into the library, they are added to the catalog, and their `category` = value is changed to the value specified by this `category` = hexnumber parameter.
`access` = permission	Either `shared` or `private`. • Specify `private` if you are using the library as one physical library. This is the default. • Specify `shared` if you are dividing the library into multiple logical libraries.
device-pathname = device-number	For device-pathname, specify the path of a drive. You must have a device-pathname entry for every drive in the library attached to this machine, and each device-pathname entry must match the Equipment Identifier value of the corresponding entry in the `mcf` file. For device-number, the device number as described in the IBM documentation. You can obtain this number by running the IBM `mtlib` utility.

Comments can appear anywhere on any line and must begin with a pound sign (#). The system ignores characters to the right of the pound sign.

The following example shows a sample `/etc/ibmatl.conf` file. Information for this file was obtained from the `mtlib` utility supplied by IBM.

```
#
# This is file: /etc/ibmatl.conf
# Set this file up according the documentation supplied by IBM.
3493a   198.174.196.50  test1
```

After the `lmcpd` daemon is running, you can use the IBM `mtlib` utility to obtain the device numbers. The following example shows output from `mtlib`.

```
# mtlib -l 3493a -D
0, 00145340 003590B1A00
1, 00145350 003590B1A01
```

The following example shows a sample parameters file for an IBM 3494 library.

```
#
# This is file: /etc/opt/SUNWsamfs/ibm50
#
name = 3493a    # From /etc/ibmatl.conf
/dev/rmt/1bn = 00145340     # From mtlib output
/dev/rmt/2bn = 00145350     # From mtlib output
access=private
category = 5
```

Note -
Configuration Example for a Shared File System on a Solaris OS Platform shows the `mcf` file that corresponds with the IBM 3494 network attached automated library parameters file that was created in this procedure. The example `mcf` file points to file `ibm50` in the `/etc/opt/SUNWsamfs` directory.

How To Configure a Sony Network Attached Automated Library Parameters File

The Sony network attached automated library operates within the SAM environment through the DZC-8000S Application Interface Library package. This software package provides the application programming interface (API) to the PetaSite Controller (PSC). For more information about the DZC-8000S interface, see the Sony PetaSite Application Interface Library DZC-8000S, which is available from Sony.

Note -
Sony network attached libraries are not supported by SAM software on an x64 hardware platform.

Note -
The information in this section applies only to Sony automated libraries that are network attached through a Sony DZC-8000S interface. Sony direct attached B9 and B35 automated libraries or Sony direct attached 8400 PetaSite automated libraries do not require a parameters file.

Before You Begin

Ensure that the following are true and the Sony network attached automated library is prepared for inclusion in a SAM environment.

- The Sony network attached automated library is operational.
- The Sony PSC configuration file is installed and working.

Follow this procedure to create a parameters file for each Sony network attached library that you want to configure.

1. Change to the `/etc/opt/SUNWsamfs` directory.
   The parameters file can be written to any directory, but the most common location is `/etc/opt/SUNWsamfs`.

> Note -
> When you created your mcf file in Configuring the File System Environment, you included the full path name to the parameters file in the mcf file. Make sure that the mcf file points to the correct location for the parameters file that you create in this procedure.

2. Create a new file and give it a name that corresponds to the library that you are configuring. For example:

```
# vi sonyfile
```

3. Type a list of keyword = value parameter lines in the Sony parameters file.

The various keyword values identify the Sony automated libraries, the drives associated with the libraries, and the host name. All keyword and value entries are case-sensitive, so type them exactly as specified in the configuration file and in the SAM mcf file.
The following table shows the keyword = value parameters that must appear in the Sony parameters file. All parameters are required.

Parameter	Meaning
userid = user-id	A number from 0 to 65535, inclusive. If you specify a number other than 0, it must be the PSC ID. The userid parameter identifies the user during initialization of the PetaSite automated library functions.
server = server-id	The host name of the server running the PSC server code.
sonydrive drive-id = path	For drive-id, the drive bin number as configured in the PSC configuration file. Include one sonydrive line for every drive defined in the mcf file. For path, the path to the drive as entered in the Equipment Identifier field of the SAM mcf file.

Comments can appear anywhere on any line, but they must begin with a pound sign (#). The system ignores characters to the right of the pound sign.

The following example shows a parameters file for a Sony network attached automated library.

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

> Note -
> Configuration Example for a Shared File System on a Solaris OS Platform shows the mcf file that corresponds with the Sony network attached automated library parameters file that was created in this procedure. The example mcf file points to file sonyfile in the /etc/opt/SUNWsamfs directory.

How To Configure a StorageTek ACSLS-Attached Automated Library Parameters File

In many respects, the way in which SAM systems interoperate with StorageTek ACSLS-attached automated libraries is very similar to the way in which they interoperate with direct attached automated libraries. However, the installation and configuration procedure of a StorageTek

ACSLS-attached automated library requires additional steps.

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

> Note -
> The SAM-QFS Manager software supports the automatic discovery and configuration of ACSLS network attached libraries. You do not need to set up the parameters file before configuring the library in SAM-QFS Manager. For more information, see the SAM-QFS Manager online help.

Before You Begin

Ensure that the following are true and the StorageTek ACSLS-attached automated library is prepared for inclusion in a SAM environment.

- The StorageTek ACSLS automated library is operational.
- The StorageTek ACSLS software package is installed and working.

Follow this procedure to create a parameters file for each StorageTek ACSLS-attached library that you want to configure. See the `stk`(7) man page for more information.

1. Change to the `/etc/opt/SUNWsamfs` directory.
   The parameters file can be written to any directory, but the most common location is `/etc/opt/SUNWsamfs`.

   > Note -
   > When you created your `mcf` file in Configuring the File System Environment, you included the full path name to the parameters file in the `mcf` file. Make sure that the `mcf` file points to the correct location for the parameters file that you create in this procedure.

2. Create a new file and give it a name that correspond to the library that you are configuring. For example:

   ```
   # vi stk50
   ```

3. Type a list of keyword = value parameter lines in the StorageTek parameters file.

The following table shows the keywords to use.

Parameter	Meaning
`access` = userid	(Optional) The user identification value used by the StorageTek software for access control. If the `access` parameter is not supplied, the access control string is a null string, indicating that there is no userid.
`hostname` = hostname	The host name of the server running the StorageTek ACSLS interface.
`portnum` = portnum	The port number used for communication between ACSLS and the SAM software. For information about the portnum argument, see the `stk`(7) man page.
`ssihost` = hostname	The name of the SAM server on the LAN that connects to the ACSLS host. Specify this directive only if you are including a multihomed SAM server in your environment. The default is the name of the local host.
`ssi_inet_port` = ssi-inet-port	The fixed port number for incoming responses and specifies the port the SSI will use for incoming ACSLS responses in a firewall environment. Specify either 0 or a value from 1024 to 65535. Setting this environmental variable to a non-zero value forces SSI to use this port for incoming ACSLS responses.
`csi_hostport` = csi-port	The port on the ACSLS server to which the StorageTek SSI daemon is to send its ACSLS requests. Specify either 0 or a value from 1024 to 65535, inclusive. Setting this variable to 0 or leaving it unset causes the system to query the port mapper on the ACSLS server.

capid = (acs = acsnum, lsm = lsmnum, cap = capnum)	The cartridge access point (CAP), in terms of the StorageTek library, to be used when the `export(1M) -f` command is specified. The `capid` description starts with an open parenthesis followed by three keyword = value pairs followed by a closing parenthesis. Use a comma (as shown), a colon, or a space to separate the keyword = value pairs. For acsnum, specify the asynchronous communications server (ACS) number for this CAP as configured in the StorageTek library. For lsmnum, specify the length subnet mask (LSM) number for this CAP as configured in the StorageTek library. For capnum, specify the CAP number for this CAP as configured in the StorageTek library.
capacity = (index = value, [index = value]...)	The capacities of the supported cartridges. Use a comma to separate the index = value pairs, and enclose the string in parentheses. For index, specify the index of the supplied `media_type` file, located in the following ACSLS directory: `/export/home/ACSSS/data/internal/mixed_media/media_types.dat`. For value, specify the capacity of the cartridge type in units of 1024 bytes. In general, supplying a capacity entry is necessary only for an index of new cartridge types or to override the supported capacity.
device-path-name = (acs = value, lsm = value, panel = value, drive = value) [shared]	The path to the device on the client. Specify one device-path-name = entry for each drive attached to this client. This parameter describes the drive within the StorageTek automated library. This description starts with an open parenthesis followed by four keyword = value pairs and a closing parenthesis. Use a comma (as shown), a colon, or a space to separate the keyword = value pairs. The `shared` keyword is optional. It specifies that the drive can be shared between two or more SAM processes from two or more hosts. For more information about implementing shared drives, see About Shared Drives or see the `stk(7)` man page. For the value arguments, use the information supplied by the ACSLS query drive command. The value specifications are: • `acs` - ACS number for the drive as configured in the StorageTek library • `lsm` - LSM number for the drive as configured in the StorageTek library • `panel` - PANEL number for the drive as configured in the StorageTek library • `drive` - DRIVE number for the drive as configured in the StorageTek library

The following example shows a parameters file for a StorageTek ACSLS-attached automated library.

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

Note -
Configuration Example for a Shared File System on a Solaris OS Platform shows the `mcf` file that corresponds with the StorageTek ACSLS-attached automated library parameters file that was created in this procedure. The example `mcf` file points to file `stk50` in the `/etc/opt/SUNWsamfs` directory.

About Shared Drives

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

The shared-drives capability enables two or more `mcf` files to define the same drive, making the drive available to multiple SAM processes. However, these multiple processes cannot share media. Each SAM process must still maintain its own set of VSNs.

The shared-drives feature can be useful, for example, if a library is attached to more than one host system in a SAM environment. The SAM processes coordinate the use of a drive and keep the drives in a library busy.

You can configure some network attached libraries to share one or all media drives between multiple SAM processes on multiple host systems. All of the StorageTek ACSLS-attached libraries support shared drives in SAM environments.

How To Create a Shared Drive for Multiple SAM Processes

- To implement one or more shared drives, specify the `shared` keyword in the parameters file for each drive that is to be shared. The placement of the `shared` keyword is specific to each manufacturer's library, so see the vendor-specific sections for more information.

How To Change the Idle Time for a Shared Drive

By default, a cartridge in a shared drive can be idle for 60 seconds before being unloaded. To change this timing, change the `shared_unload` value in the `defaults.conf` file to the new value (in seconds).

Checking the Drive Order in Libraries

If your automated library contains more than one drive, the order of the drives in the `mcf` file must be the same as the order of the drives displayed by the automated library's controller. The drive that the library controller identifies as the first drive must be the first drive entry for that library in the `mcf` file, and so on. This order can be different from the order of the devices as reported in the `/var/adm/messages` file.

The following sections describe procedures for checking the drive order. The procedure varies depending on whether your automated library has a front panel, whether it has tape or magneto-optical drives, and whether it is direct attached or network attached. Each procedure has two phases:

- Mapping the library drives to SCSI target IDs
- Mapping the SCSI target IDs to remote tape devices

How to Check the Drive Order of Libraries With a Front Panel

Some libraries have a panel that display the drive information.
The following procedure is a general plan. The actual steps depend on your specific library product. Consult your vendor documentation for information about drive identification and target identification.

1. Verify the order of the drives, according to the vendor's documentation.
2. In the front panel, check each drive's SCSI target ID or World Wide Name (WWN).
3. Record the order in which each drive and drive target is reported.
4. In the `mcf` file, make sure that the order of the drive targets is in the same order in which the drives are displayed by the automated library's controller.
5. If you made any changes, verify the `mcf` file and test the drives. Then propagate the change to the rest of the system. For information about propagating `mcf` file changes, see the Sun QFS File System Configuration and Administration Guide.

To determine whether the drives become active when loaded with a cartridge, you can visually inspect the drives or use the `samu`(1M) utility's `r` display.

How to Check the Drive Order of a Tape Library Without a Front Panel

1. Stop the SAM software so that no drives are used during the procedure.
2. Get a listing of devices in /dev/rmt/, as shown in the following example:

```
# ls -l /dev/rmt/?
lrwxrwxrwx 1 root root 42 Jan 10 2000 /dev/rmt/0 ->
../../devices/pci@1f,4000/scsi@2,1/st@2,0:
lrwxrwxrwx 1 root root 42 Jan 10 2000 /dev/rmt/1 ->
../../devices/pci@1f,4000/scsi@4,1/st@5,0:
lrwxrwxrwx 1 root root 42 Jan 10 2000 /dev/rmt/2 ->
../../devices/pci@1f,4000/scsi@4,1/st@6,0:
lrwxrwxrwx 1 root other 40 Dec 13 2000/dev/rmt/3 ->
../../devices/pci@1f,4000/scsi@4/st@1,0:
lrwxrwxrwx 1 root root 40 Jun 20 2001 /dev/rmt/4 ->
../../devices/pci@1f,4000/scsi@4/st@2,0:
lrwxrwxrwx 1 root root 40 Jun 20 2001 /dev/rmt/5 ->
../../devices/pci@1f,4000/scsi@4/st@3,0:
lrwxrwxrwx 1 root root 40 Jun 20 2001 /dev/rmt/6 ->
../../devices/pci@1f,4000/scsi@4/st@4,0:
lrwxrwxrwx 1 root root 40 Sep 14 2001 /dev/rmt/7 ->
../../devices/pci@1f,4000/scsi@2/st@2,0:
lrwxrwxrwx 1 root root 40 Sep 14 2001 /dev/rmt/8 ->
../../devices/pci@1f,4000/scsi@2/st@3,0:
lrwxrwxrwx 1 root root 40 Sep 14 2001 /dev/rmt/9 ->
../../devices/pci@1f,4000/scsi@2/st@4,0:
```

3. Load a tape into the library's Drive 1. Make sure the other drives are empty.

```
samload
```

4. Run the following command with each /dev/rmt/ entry until the command returns information about the drive and tape position:

```
mt -f /dev/rmt/<x> status
```

The /dev/rmt/# entry that returns information corresponds to the library's Drive 1. The following example shows `mt`(1) command output that indicates that 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
```

5. Repeat the steps for each library drive.
Create a table that shows which library drive corresponds to which /dev/rmt/ entry:

```
   drive 1 = /dev/rmt/4 -> ../../devices/pci@1f,4000/scsi@4/st@2,0:
   drive 2 = /dev/rmt/7 -> ../../devices/pci@1f,4000/scsi@2/st@2,0:
   ...
```

6. In the `mcf` file, make sure that the drives are listed in the order in which the drives are displayed by the automated library's controller. In this case, the `mcf` file starts with the following items:

```
   # Equipment         Eq   Eq    Family   Device   Additional
   # Identifier        Ord Type    Set     State    Parameters
   #-----------        --- ----   ------   ------   ----------
   /dev/rmt/4          31   li    ibm3580   on
   /dev/rmt/7          32   li    ibm3580   on
    ...
```

7. If you made any changes, verify the `mcf` file and test the drives. Then propagate the change to the rest of the system. For information about propagating `mcf` file changes, see the Sun QFS File System Configuration and Administration Guide.

How to Check the Drive Order of Magneto-Optical Libraries Without a Front Panel

1. Stop the SAM software so that no drives are used during the procedure.
2. Get a listing of devices in /dev/samst/, as shown in the following example:

```
# ls -l /dev/samst/?
```

3. Load a magneto-optical cartridge manually through the library front panel into the library's Drive 1. Make sure the other drives are empty.
4. Run the following command with each /dev/samst/ entry until the command returns information about the drive and tape position:

```
dd if=</dev/samst/x> bs=2k iseek=3374 of=/tmp/foo count=10
```

The /dev/samst/ entry that returns information corresponds to the library's Drive 1. The following example shows a status message that indicates that an optical cartridge is in the selected device.

```
# dd if=/dev/samst/c0t3u0 bs=2k iseek=3374 of=/tmp/junk count=10
10+0 records in
10+0 records out
```

5. Repeat the steps for each library drive.
Create a table that shows which library drive corresponds to which /dev/samst/ entry:

```
   drive 1 = /dev/samst/4 -> ../../devices/pci@1f,4000/scsi@4/st@2,0:
   drive 2 = /dev/samst/7 -> ../../devices/pci@1f,4000/scsi@2/st@2,0:
   ...
```

6. In the `mcf` file, make sure that the drives are listed in the order in which the drives are displayed by the automated library's controller. In this case, the `mcf` file starts with the following items:

```
   # Equipment         Eq   Eq    Family   Device   Additional
   # Identifier        Ord Type    Set     State    Parameters
   #-----------        --- ----   ------   ------   ----------
   /dev/samst/4          31   li    ibm3580   on
   /dev/samst/7          32   li    ibm3580   on
    ...
```

7. If you made any changes, verify the `mcf` file and test the drives. Then propagate the change to the rest of the system. For information about propagating `mcf` file changes, see the Sun QFS File System Configuration and Administration Guide.

How to Check the Drive Order of Network Attached Libraries

1. Stop the SAM software so that no drives are used during the procedure.
2. Get a listing of devices in /dev/rmt/, as shown in the following example:

```
   # ls -l /dev/rmt/*[0-9] | awk '{print $9, $10, $11}'
   /dev/rmt/0 -> /devices/pci@8,700000/SUNW,qlc@4,1/fp@0,0/st@w500104f0006041f0,0:
   /dev/rmt/1 -> /devices/pci@8,700000/SUNW,qlc@4,1/fp@0,0/st@w500104f0006041f3,0:
   /dev/rmt/2 -> /devices/pci@8,700000/SUNW,qlc@4,1/fp@0,0/st@w500104f00043cbb8,0:
   /dev/rmt/3 -> /devices/pci@8,700000/SUNW,qlc@5,1/fp@0,0/st@w500104f0006041ea,0:
   /dev/rmt/4 -> /devices/pci@8,700000/SUNW,qlc@5,1/fp@0,0/st@w500104f0006041ed,0:
   /dev/rmt/5 -> /devices/pci@8,700000/SUNW,qlc@4/fp@0,0/st@w500104f00060420e,0:
   /dev/rmt/6 -> /devices/pci@8,700000/SUNW,qlc@4/fp@0,0/st@w500104f000604211,0:
   /dev/rmt/7 -> /devices/pci@8,700000/SUNW,qlc@4/fp@0,0/st@w500104f000604214,0:
   /dev/rmt/8 -> /devices/pci@8,700000/SUNW,qlc@5/fp@0,0/st@w500104f000604208,0:
   /dev/rmt/9 -> /devices/pci@8,700000/SUNW,qlc@5/fp@0,0/st@w500104f00060420b,0:
```

Next we can use luxadm output, along with output from an ACSLS display command and associate the serial numbers of each drive with the physical location in the library.

3. For each device, use the `luxadm display` command to show its serial number:

```
# luxadm display /dev/rmt/<x>
```

4. Use the ACSLS `display` to display the drive identifier for each serial number.

```
ACSSA> display drive * -f serial_num
2007-10-11 10:49:12              Display Drive
Acs  Lsm  Panel  Drive  Serial_num
0    2    10     12      331000049255
0    2    10     13      331002044567
0    2    10     14      331002057108
0    2    10     15      331002042417
0    2    10     16      331002031352
0    2    10     17      HU92K00200
0    2    10     18      HU92K00208
0    3    10     10      1200019405
0    3    10     11      1200019442
0    3    10     12      1110150718
```

5. Create a table to show the relationships among the identifiers:

```
 Device          SSN              Drive Identifier
/dev/rmt/0 -> 331000049255 -> (acs=0, lsm=2, panel=10, drive=12)
/dev/rmt/1 -> 331002044567 -> (acs=0, lsm=2, panel=10, drive=13)
/dev/rmt/2 -> 331002057108 -> (acs=0, lsm=2, panel=10, drive=14)
```

6. In the `mcf` file, verify that the order of the drives is in the same order as the table.
7. If you made any changes, verify the `mcf` file and test the drives. Then propagate the change to the rest of the system. For information about propagating `mcf` file changes, see the Sun QFS File System Configuration and Administration Guide.

Populating the Catalog

After you mount a file system, the SAM-QFS software creates catalogs for each automated library configured in the `mcf` file. However, if you have a network attached automated library, you need to populate the library's catalog. The appropriate method depends on the number of volumes you include in the catalog.

How to Populate an Automated Library With Many Volumes

Use this procedure for ADIC/Grau, Sony network attached, StorageTek ACSLS-Attached, and IBM 3494 automated libraries.

Note the following when creating the input file:

- The file has four fields in each row. Each row identifies a volume. For each volume, specify the slot number, the VSN, the bar code, and the media type.

> Note -
> The slot position of a tape in a network attached automated library has no relationship to the slot number of the volume in a Sun Storage Archive Manager (SAM-QFS) library catalog.

- Use a space character or a tab character to separate the fields in this file.
- If a VSN contains one or more space characters, enclose the VSN name in quotation marks (" ").

1. Create an input file that contains the slot number, the volume's VSN, the barcode number, and the media type.
The following example shows the sample file `input_vsns`.

```
0 TAPE01  "TAPE 01" lt
1 TAPE02  TAPE02  lt
2 TAPE03  TAPE03  lt
```

2. Use the `build_cat`(1M) command to create the catalog.

```
build_cat <input-file> <catalog-file>
```

Argument	Content
input-file	The name of an input file. Typically, this is a file containing a list of VSNs.
catalog-file	The full path to the library catalog. By default, the Sun Storage Archive Manager software creates a catalog and writes it to `/var/opt/SUNWsamfs/catalog/`family-set-name, where family-set-name is derived from the `mcf` file entry for this automated library. Alternatively, if you have specified a catalog name in the Additional Parameters field of the `mcf` file, use that catalog file name for catalog-file.

For example, you might specify the following `build_cat`(1M) command:

```
# build_cat input_vsns /var/opt/SUNWsamfs/catalog/grau50
```

How to Populate an Automated Library With a Small Number of Volumes

Use this procedure for ADIC/Grau, Sony network attached, StorageTek ACSLS-Attached, and IBM 3494 automated libraries.

Perform this procedure for each cartridge that you want to include in the catalog. The cartridge must be physically present in the automated library. If the cartridge is not present, the entry is recorded in the historian, which keeps track of cartridges exported from an automated library or a manually mounted device and you see an error message, as described in Example 3 - Message Generated After a `samimport`(1M) Command. For more information about the historian, see Tracking Exported Media-The Historian.

- Use the `samimport`(1M) command to import catalog entries into the default catalog.

```
samimport -v <VSN> <eq>
```

Argument	Content
VSN	Specify the VSN identifier for a volume. If a VSN name contains one or more space characters, enclose the VSN name in quotation marks (" ").
eq	Specify the Equipment Ordinal as specified for the device in the `mcf` file.

For example:

```
# samimport -v TAPE01 50
```

How to Populate an IBM 3494 Automated Library

Use this procedure only if you are using an IBM 3494 library as one physical library, that is, if `access=private` is specified in the `mcf` file. Do not use this procedure if you divided the library into multiple logical libraries.

> **Note -**
> If you have an IBM 3494 library that is divided into multiple logical libraries, that is, `access=shared` is specified in the IBM 3494 parameters file, use one of the previous methods to populate the catalog: How to Populate an Automated Library With Many Volumes or How to Populate an Automated Library With a Small Number of Volumes.

- Insert the media cartridge into the mail slot.
  The library automatically builds a catalog that includes the media cartridge.

How to Populate a StorageTek ACSLS-Attached Library Quickly

This procedure is an alternative, faster method of populating a library catalog than the methods described in How to Populate an Automated Library With Many Volumes or How to Populate an Automated Library With a Small Number of Volumes.

- Use the `samimport`(1M) command with its `-c` and `-s` options to import from a pool of VSNs.
  For more information, see the `samimport`(1M) man page.

StorageTek ACSLS-Attached Automated Libraries: Common Problems and Error Messages

If errors exist in the configuration files for a StorageTek ACSLS-attached automated library, the system generates several error messages. The following examples show common problems and the messages that the system generates.

Example 1 - Errors From a StorageTek ACSLS Parameters File

```
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 your StorageTek parameters file for syntax errors.
Each line must begin with a keyword or a comment.
For more information about the StorageTek parameters file, see the `stk`(7) man page.

Example 2 - Error Messages from a StorageTek ACSLS Library

The following example shows drives that are frozen in the initializing state.

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

The following example shows the `samu`(1M) utility's `:r` display.

```
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 frozen in an initializing state or that do not initialize indicate a configuration error.

- Verify that ACSLS software is running.
- Verify the host name.
- Determine whether you can you reach the host using the `ping`(1M) command.
- Check the `portnum` specification in the StorageTek parameters file. In ACSLS 5.3, for example, the default port number, 50004, is used for a different application. Try a higher port number, such as 50014.

Example 3 - Message Generated After a ***samimport***(1M) Command

Example 3 shows messages generated after the samimport(1M) command was used to import a VSN to the library catalog, but the VSN was not in the StorageTek automated library. The cartridge must be present in the ACSLS-managed automated library before the samimport(1M) command can be successful.

```
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
```

> Note -
> The sam-stkd daemon uses the ssi.sh script to ensure that a copy of the SSI daemon, ssi_so, is running. If ssi_so exits, the daemon starts another. If your site has its own version of ssi.sh, modify it to wait for a SIGTERM signal and then exit. The daemon sends a SIGTERM signal to stop the process. The /opt/SUNWsamfs/examples/ssi.sh file contains an example ssi.sh script. The system copies the ssi.sh script to /etc/opt/SUNWsamfs/scripts/ssi.sh during installation if one does not already exist.

Managing Automated Libraries and Manually Loaded Drives

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

This page describes aspects of using automated libraries and manually loaded drives in a Sun Storage Archive Manager (SAM-QFS) environment. In addition, this page describes the operator-oriented load notification facility that alerts an operator when a requested volume is not in a library.

> Note -
> The SAM-QFS software interoperates with automated libraries from many manufacturers. Contact Sun Customer Support for information pertinent to library model numbers, firmware levels, and other compatibility information.

Vendor-Specific Library Operational Procedures

Certain operations for some automated libraries may differ from those described in this chapter. To determine whether your automated library has additional vendor-specific operating instructions in a SAM-QFS environment, see the Managing Vendor-Specific Libraries page.

How To Start Removable Media Operations

Typically, removable media operations commence when a file system is mounted.

- To start removable media operations manually, without mounting any file systems, issue the following command:

  ```
  # samd start
  ```

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

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

How To Stop Removable Media Operations

You can stop removable media operations and leave the file system mounted. You might do this, for example, if you want to manipulate cartridges manually.
Use the following procedure to stop operations:

#Issue the `idle` command to enable the archiver, stager, and other processes to complete current tasks. You can also idle drives by using the `samu`(1M) operator utility or by using SAM-QFS Manager.

```
samcmd aridle
samcmd stidle
```

Monitor the tape drive activity with the samcmd r command:

```
samcmd r
```

Wait to all the tape drives have stopped. Then unload the tape drives:

```
samcmd unload <eq>
```

For eq, provide the equipment number of the drive being unloaded as defined in the `mcf` file.

1. To unload multiple drives, issue a `samcmd unload` for each drive.
2. When the drive is empty, issue the following command:

   ```
   samd stop
   ```

   > Note -
   > Failing to follow the above procedures could result in tape media issues.
   > You must issue the `samd`(1M) `unload` command for each idle tape drive. If you attempt to restart an idle drive without unloading it, unpredictable events occur when archiving, staging, and other activities are resumed.

When you restart operations, pending stages are reissued and archiving is resumed.

How To Turn On an Automated Library

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

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

- Use the following command to start an automated library:

  ```
  samcmd on <eq>
  ```

  For eq, specify the equipment number of the automated library as defined in the `mcf` file.
  You can also perform this task by using `samu`(1M) or SAM-QFS Manager.

How To Turn Off an Automated Library

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

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

- Use the following command to turn off an automated library:

```
samcmd off <eq>
```

For eq, specify the equipment number of the automated library being addressed as defined in the `mcf` file.
You can also perform this task by using `samu`(1M) or SAM-QFS Manager.

How To Load a Cartridge Manually

When a cartridge is loaded, it moves from a storage slot to a drive and it is made ready to receive data. A cartridge is loaded automatically when a volume serial name (VSN) is requested for archiving or staging. You can load a cartridge at any time by issuing one of the following commands. You might do this, for example, during a disaster recovery operation or to analyze a tape.

- ```
  samcmd load <eq>:<slot>[:<partition>]
  ```
- ```
  samcmd load <media-type>.<vsn>
  ```

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

You can also perform this task by using `samu`(1M) or SAM-QFS Manager.
When you load a cartridge manually, it is loaded in the next available drive in the library. If you want to make a drive unavailable for this purpose, use the `samu`(1M) utility's `:unavail` command or change the state of the device using SAM-QFS Manager.

> Note -
> SAM-QFS does not support mixed media in direct attached libraries. If the library is partitioned, each partition must contain only one media type.

How To Unload a Cartridge Manually

When a cartridge is unloaded, it is removed from a drive. Unloading occurs automatically when a volume is no longer needed. You can unload a drive at any time, even if the drive is in `unavail` status, using the following command.

```
samcmd unload <eq>
```

For eq, specify the equipment number of the drive as defined in the `mcf` file.
You can also perform this task by using `samu`(1M) or SAM-QFS Manager.

Labeling and Unlabeling Cartridges

If you have stand-alone tape or optical devices, or if your automated library has no barcode reader, you must label cartridges as described in this section. To label cartridges, use the `tplabel`(1M) command for tapes or use the `odlabel`(1M) command for optical disks. These commands create a cartridge label that the SAM-QFS software can read.

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

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

```
labels = barcodes_low
```

When the software loads a barcoded cartridge for a write operation, it writes a label on the cartridge before the write begins. The cartridge must be write enabled, be unlabeled, and have a readable barcode.

How To Label or Relabel a Tape

Caution -
Labeling and relabeling a cartridge makes the data currently on the cartridge inaccessible to any software. Relabel a cartridge only if you are certain that you do not need the data that is stored on the cartridge.

- To label a new tape, use the following `tplabel`(1M) command.

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

- To relabel an existing tape, use the following `tplabel`(1M) command.

```
tplabel -old <vsn> -vsn <vsn> <eq>:<slot>
```

Argument	Meaning
vsn	Volume serial name. If you are relabeling, the new VSN can be identical to the old VSN.
eq	The equipment number of the drive as defined in the `mcf` file.
slot	The number of the tape's storage slot as recognized in the library catalog. This argument is not applicable for manually loaded drives.

After the command to label or relabel a tape is issued, the tape is loaded and positioned, and the tape label is written.

You can also perform this task by using SAM-QFS Manager.

Example

```
# tplabel -vsn TAPE01 -new 50:0
```

How To Label or Relabel an Optical Disk

- To label a new optical disk, use the following `odlabel`(1M) command.

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

- To relabel an existing optical disk, use the following `odlabel`(1M) command.

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

Argument	Meaning
vsn	Volume serial name. If you are relabeling, the new VSN can be identical to the old VSN.
eq	The equipment number of the drive as defined in the `mcf` file.
slot	The number of the disk's storage slot as recognized in the library catalog. This argument is not applicable to manually loaded drives.
partition	A side of a magneto-optical disk. The partition value must be 1 or 2. This argument is not applicable to tape cartridges.

After the command to label or relabel an optical disk is issued, the optical disk is loaded and positioned, and the optical disk label is written.

You can also perform this task by using SAM-QFS Manager.

Example

```
# odlabel -vsn OPTIC01 -new 30:1:1
```

How To Audit a Volume

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

- Use the following command to update the amount of remaining space:

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

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

For more information, see the auditslot(1M) man page.
You can also perform this task by using the `samu`(1M) utility's `:audit` command or by using SAM-QFS Manager.

How To Audit a Direct Attached Automated Library

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

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

- To perform a full audit on a direct attached automated library, use the following command:

```
samcmd audit <eq>
```

  For eq, specify the equipment number of the automated library as defined in the `mcf` file.
  You can also perform this task by using the `samu`(1M) utility's `:audit` command or by using SAM-QFS Manager.

Using a Cleaning Cartridge

The SAM-QFS environment supports the use of cleaning tapes if cleaning tapes are supported by the hardware. If a tape drive requests cleaning, the system automatically loads a cleaning tape.

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

Cleaning practices differ from manufacturer to manufacturer. See Managing Vendor-Specific Libraries to determine whether specialized procedures are recommended for your equipment.

How To Use a Cleaning Cartridge With a Barcode

If the cleaning cartridge is barcoded, you can import it using the `samimport`(1M) command. This command moves the cartridge from the mailbox to a storage slot and updates the library catalog. In addition, the cleaning media flag is set, and the access count is set to the appropriate number of cleaning cycles, based on the media type. Each time the cartridge is used to clean a drive, the access count is decremented.

1. Make sure that the cleaning cartridge has a barcode of `CLEAN` or starts with the letters `CLN`.
2. Use the following command to import the cleaning cartridge into the automated library.

```
samimport <eq>
```

  For eq, specify the equipment number of the automated library as defined in the `mcf` file.
  You can also perform this task by using `samu`(1M) or SAM-QFS Manager.

How To Use a Cleaning Cartridge Without a Barcode

If the cartridge is not barcoded, it is not identified as a cleaning cartridge. After you import the cartridge, you must identify it.

1. Import the cartridge into the automated library.

```
samimport <eq>
```

For eq, specify the equipment number of the automated library as defined in the `mcf` file.

2. Change the type to a cleaning cartridge.

```
chmed +C <eq>:<slot>
```

For eq, specify the equipment number of the automated library.
For slot, specify the slot in which the cleaning cartridge has been loaded.
In the following example, the automated library is equipment number 50 and the cleaning cartridge is in slot 77:

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

3. Set the cleaning cycle count.

```
 chmed -count <count-number> <eq>:<slot>
```

For eq, specify the equipment number of the automated library.
For slot, specify the slot in which the cleaning cartridge has been loaded.
The following example command sets the cleaning count on the cartridge to 20.

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

How To Reset the Number of Cleaning Cycles

Cleaning cartridges are useful for a limited number of cleaning cycles. The SAM-QFS system ejects the cartridge when the number of remaining cycles equals zero. Each time a cleaning tape is imported, the cleaning cycle is reset to the highest number of cycles for that type of tape. For example, a DLT cleaning tape has 20 cycles and an Exabyte cleaning tape has 10 cycles. You can view the number of remaining cycles with the `samu`(1M) utility's `:v` display or by using SAM-QFS Manager.

If automatic cleaning is available but all cleaning cartridges in the automated library have a cleaning cycle count of zero, the drive state is set to `off` and a message is issued in the SAM-QFS log.

- To reset the cycles on a cleaning tape:

```
chmed -count <count> <media-type>.<vsn>
```

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

How To Limit the Number of Cleaning Cycles

Certain drive errors can result in the repeated loading of cleaning cartridges until all cleaning cycles are consumed.

- To limit the number of cleaning cycles on cleaning cartridges:

```
chmed -count <count-number> <eq>:<slot>
```

For eq, specify the equipment number of the automated library.
For slot, specify the slot in which the cleaning cartridge has been loaded.

For example, the following command sets the cleaning cycle count to 20 for the cleaning cartridge in slot 77 of the automated library with the equipment number of 50.

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

How To Clean a Tape Drive Automatically

Beginning with the Sun Storage Archive Manager 4.4 version, the default setting for software-initiated tape drive cleaning is `off`. You can enable automatic cleaning in one of the following ways:

- Use the media changer's auto-cleaning feature, which might require specific placement cleaning cartridges. See the manufacturer's documentation for directions.
- Enable the SAM-QFS auto-cleaning feature:

1. Disable the media changer's cleaning feature, according to the manufacturer's documentation.
2. Edit the `defaults.conf` file to add the following line:

```
tapeclean = all autoclean on logsense on
```

The `logsense` option prevents a drive from using expired cleaning media. If you prefer to use only sense data for determining the status of cleaning media, use the following line:

```
tapeclean = all autoclean on logsense off
```

Note -

When using the auto-cleaning feature with a library with more than two drives, use at least two cleaning cartridges for each catalog. If not enough cleaning cartridges are available, any drive that requires cleaning is put into a `DOWN` state.

How To Clean a Tape Drive Manually

When automatic cleaning is not available and the system uses barcodes, you can request that a drive be cleaned at any time.

- Use the following command:

```
cleandrive <eq>
```

For eq, specify the equipment number of the automated library as defined in the `mcf` file.
This drive is loaded with the cleaning cartridge.

How To Clear Media Errors

> **Note -**
> Removing the error flag can cause problems. If you are not certain of what caused the error and that the flag can be removed safely, contact Sun Technical Support and do not use this procedure.

When a hardware or software error is encountered on a cartridge, the SAM-QFS system sets the `media error` flag in the VSN catalog. The `media error` flag is displayed in the `samu`(1M) utility's `v` display and in SAM-QFS Manager.
You can clear the error to reset the flag and you can then attempt to use the cartridge.

1. To clear the `media error` flag on a cartridge:

```
chmed -E <media-type>.<vsn>
```

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

2. Update the library catalog with the space remaining information.

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

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

For more information, see the `auditslot`(1M) man page.
You can also perform this task by using the `samu`(1M) utility's `:audit` command or SAM-QFS Manager.

How To Remove a Stuck Cartridge From a Drive

1. Use the `samcmd off` command to turn off the drive in the automated library.

```
samcmd off <eq>
```

For eq, specify the equipment number of the drive as defined in the `mcf` file.
You can also perform this step by using `samu`(1M) or SAM-QFS Manager.

2. Use the `samcmd off` command to turn off the automated library.

```
samcmd off <eq>
```

For eq, specify the equipment number of the library as defined in the `mcf` file.
You can also perform this step by using `samu`(1M) or SAM-QFS Manager.

3. Physically remove the cartridge from the drive.
Make sure you do not damage either the cartridge or the drive.
4. Use the `samcmd on` command to turn on the automated library and the drive.
Issue this command once for the drive and once for the library.

```
samcmd on <eq>
```

For eq, specify the equipment number of the library and then of the drive as defined in the `mcf` file.
If the automated library performs an audit when it is turned on, you are done.

5. Follow these steps if the automated library does not perform an audit:
    a. Put the cartridge back into its storage slot.
    b. Use the `chmed`(1M) command to adjust the library catalog to set the occupied flag for the damaged tape.

    ```
    chmed +o <eq>:<slot>
    ```

Argument	Meaning
eq	The equipment number of the automated library or drive as defined in the `mcf` file.
slot	The number of the storage slot in the library as recognized in the library catalog. This argument is not applicable for manually loaded drives.

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

Catalog Operations, Importing Cartridges, and Exporting Cartridges

The physical addition (import) of cartridges to and removal (export) of cartridges from an automated library enables you to perform several functions, including the following:

- Replace cartridges.
- Relocate cartridges to off-site storage to use later for disaster recovery purposes. You can use the `-I` option on the `chmed`(1M) command to specify additional information such as the storage location of the cartridge.

When you import and export cartridges, you also update the library catalog.

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

The SAM-QFS environment treats catalogs differently depending on how the automated library is attached to the server, as follows:

- If the automated library is direct attached, the library catalog is a one-to-one mapping between library catalog entries and physical slots in the automated library. The first entry in the library catalog is for the first slot in the automated library. When a cartridge is needed, the system consults the library catalog to determine which slot contains the VSN, and it issues a command to load the cartridge from that slot into a drive.

    > Note -
    > SAM-QFS does not support mixed media in direct attached libraries. If the library is partitioned, each partition must contain only one media type.

- If the automated library is network attached, the library catalog is not a direct mapping to the slots. It is a list of the VSNs known to be present in the automated library. When a cartridge is requested, the system sends a request to the vendor's software to load the VSN into a drive. The vendor's software locates the VSN's storage slot.

Each automated library handles cartridge import and export differently due to system characteristics and the vendor-supplied software. For example, on the ACL 4/52 library, you need to issue a `move` command to move cartridges into the import/export unit before exporting cartridges from the automated library.

Network attached automated libraries import and export cartridges using their own utilities, so the `samimport`(1M) and `samexport`(1M) commands update only the library catalog entries used by the SAM-QFS systems. If you have a network attached library, see Managing Vendor-Specific Libraries for information about importing and exporting cartridges.

Tracking Exported Media - The Historian

The SAM-QFS historian keeps track of cartridges exported from an automated library or a manually mounted device. The historian acts like a virtual library, but it has no defined hardware devices. It has the following similarities to an automated library:

- Is configured in the mcf file
  You can configure the historian in the mcf file by using a device type of hy. If you do not configure the historian in the mcf file, it is created as follows:

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

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

- Has a catalog that records entries for all cartridges associated with it
  The historian library catalog is initialized with 32 entries when the historian first starts. Make sure that the catalog resides on a file system large enough to hold the entire catalog. Your site might want to track existing SAM-QFS cartridges that have been exported from the library. In this case, you need to build a historian catalog from the existing cartridges as described in the build_cat(1M) man page.
  The following two configuration directives in the defaults.conf file affect the behavior of the historian:
  - The exported_media = unavailable directive flags any cartridges exported from an automated library as unavailable to the historian. Requests for these cartridges generate EIO errors.
  - The attended = no directive declares to the historian that no operator is available to handle load requests. Requests to load cartridges that are not already loaded generate EIO errors.
    For more configuration information, see the historian(7) and defaults.conf(4) man pages.
- Can import and export cartridges
  Importing and exporting practices differ from manufacturer to manufacturer. See Managing Vendor-Specific Libraries to determine if specialized procedures are recommended for your equipment.
- Is displayed in SAM-QFS Manager as another automated library

About Importing and Exporting From an Automated Library

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

How To Import a Cartridge From a Library With a Mailbox

1. Open the mailbox using the manufacturer's suggested operation, usually by a button near the mailbox. Sometimes the mailbox is a one-slot mailbox referred to as a mail slot in the vendor's documentation.
2. Manually place the cartridge in the mailbox.
3. Close the mailbox.
4. Use the samimport(1M) command to import the cartridge.

   ```
   samimport <eq>
   ```

   For eq, specify the equipment number of the library as defined in the mcf file.
   The system moves the cartridge from the mailbox to a storage slot and updates the library catalog for the cartridge.
   You can also perform this task by using samu(1M) or SAM-QFS Manager.

How To Export a Cartridge From a Library With a Mailbox

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

   - ```
     samexport <eq>:<slot>
     ```
   - ```
     samexport <media-type>.<vsn>
     ```

Argument	Meaning

eq	The equipment number of the automated library as defined in the `mcf` file.
slot	The number of the storage slot in the automated library as recognized in the library catalog.
media-type	The media type of the cartridge. For a list of valid media types, see the `mcf(4)` man page.
vsn	The volume serial name (VSN) assigned to the volume.

You can also perform this step by using samu(1M) or SAM-QFS Manager.

1. Open the mailbox or mail slot using the manufacturer's instructions, usually by pressing a button near the mailbox.

How To Import a Cartridge From a Library Without a Mailbox

1. Use the following command:

```
samcmd unload <eq>
```

For eq, specify the equipment number of the library as defined in the `mcf` file.

2. Wait until the system completes its current task, sets the status to `off`, and transfers the current active catalog to the historian.
3. Unlock and open the door to the automated library.
4. Load cartridges into the available slots.
5. Close and lock the door to the automated library.

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

How To Export a Cartridge From a Library Without a Mailbox

1. Use the following command:

```
samcmd unload <eq>
```

For eq, specify the equipment number of the library being addressed as defined in the `mcf` file.

2. Wait until the system completes its current task, sets the status to `off`, and transfers the current active catalog to the historian.
3. Unlock and open the door to the automated library.
4. Remove the cartridges from their respective slots.
5. Close and lock the door to the automated library.

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

How To Enable Load Notification

The SAM-QFS software requests cartridges to be loaded regularly to satisfy archiving and staging needs. If the request is for a cartridge that resides inside a library, the request is handled automatically. If the request is for a cartridge that resides outside the library, operator action is required. If enabled, the `load_notify.sh` script sends email when a cartridge needs to be obtained from outside the library.

1. Become superuser.
2. Copy the load notification script from its installed location to its operable location. For example:

```
# cp /opt/SUNWsamfs/examples/load_notify.sh /etc/opt/SUNWsamfs/scripts/load_notify.sh
```

3. Use `more(1)` or another command to examine the `defaults.conf` file. Make sure that the following default directives are in the file and have not been changed.

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

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

Using Drives With Encryption Capability

If you are archiving files to drives with encryption capability, plan your archive operations according to the following special considerations:

- Do not mix non-encrypted and encryption-capable drives in a library.
- After a drive has encryption enabled, it cannot be disabled.
- Do not mix encrypted and non-encrypted files on a tape.
- An encrypted drive cannot append to a tape that contains non-encrypted data.
- An encryption-enabled drive can read non-encrypted data.

Manually Loaded Drive Operations

This section describes operations that differ if you have a manually loaded, stand-alone drive rather than an automated library. Each manually loaded drive has its own one-slot library catalog.

How To Load a Cartridge

- To load a cartridge into a manually loaded device, place the cartridge in the drive according to the manufacturer's instructions.
  The SAM-QFS system recognizes that the cartridge is loaded, reads the label, and updates the one-slot catalog. No further action is necessary.

How To Unload a Cartridge

- Use `samcmd idle` command to idle the drive.

```
samcmd idle <eq>
```

  For eq, specify the equipment number of the drive as defined in the `mcf` file.

This command ensures that no archive or stage processes are active. The drive switches from `idle` to `off` when all I/O activity is complete, and the tape ejects.
If the cartridge is a tape, the tape rewinds and is ready to be removed. An optical disk ejects automatically. See the manufacturer's instructions for removing the specific cartridge.
You can also perform this task by using `samu`(1M) or SAM-QFS Manager.

How To View a Library Catalog

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

```
:v <eq>
```

  For eq, specify the equipment number of the library as defined in the `mcf` file.

Managing Vendor-Specific Libraries

You can include libraries from many different manufacturers in a Sun Storage Archive Manager (SAM-QFS) environment. For most libraries, use the operational procedures described in Managing Automated Libraries and Manually Loaded Drives. The following libraries, however, have vendor-specific operational procedures:

- ADIC/Grau Automated Libraries
- IBM 3584 UltraScalable Tape Libraries
- IBM 3494 Libraries
- Sony Direct Attached 8400 PetaSite Automated Libraries
- Sony Network Attached Automated Libraries
- StorageTek ACSLS-Attached Automated Libraries

> Note -
> Consult your Oracle sales representative or your authorized service provider for information about library model numbers, firmware levels, and other compatibility information.

ADIC/Grau Automated Libraries

> Note -
> ADIC/Grau network attached libraries are not supported by SAM-QFS software on an x64 hardware platform.

The following sections describe aspects of this library's operations that differ from the procedures described in Managing Automated Libraries and Manually Loaded Drives.

Because you use vendor-supplied utilities to physically add and remove cartridges in the ADIC/Grau automated library, the SAM-QFS interfaces (`samimport`(1M), `samexport`(1M), and SAM-QFS Manager) affect only the library catalog.

How To Import a Cartridge

1. Use ADIC/Grau commands to physically move the cartridge into the library.
2. Use the `samimport`(1M) command to update the library catalog.

```
samimport -v <volser> <eq>
```

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

How To Export a Cartridge

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

```
samexport <eq>:<slot>
samexport <media-type>.<vsn>
```

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

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

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

IBM 3584 UltraScalable Tape Libraries

> Note -
> IBM 3584 UltraScalable libraries are not supported by the SAM-QFS software on an x64 hardware platform.

The following sections describe aspects of this library's operations that differ from the procedures described in Managing Automated Libraries and Manually Loaded Drives.

Importing Cartridges

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

Cleaning Drives

Disable automatic cleaning and enable hosted cleaning. This process is described in the IBM 3584 UltraScalable Tape Library Planning and Operator Guide, IBM publication GA32-0408-01. This process is also described in the `ibm3584`(7) man page.

Partitioning

This library accommodates several tape drives. If you use multiple drives, you can divide this one physical library into two, three, or four logical libraries. If you do divide your library into two or more logical libraries, be sure these logical libraries are operating properly before you add the IBM 3584 library to the SAM-QFS environment.

When a cartridge is exported from a partitioned library, only the logical library from which it was exported can get access to that drawer slot. If the cartridge is removed and reinserted manually, any logical library can get access to the drawer slot.

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

How To Remove a Cartridge From a Logical Library

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

IBM 3494 Libraries

> Note -
> IBM 3494 network attached libraries are not supported by the SAM-QFS software on an x64 hardware platform.

The following sections describe aspects of this library's operations that differ from the procedures described in Managing Automated Libraries and Manually Loaded Drives.

How To Import a Cartridge

1. Place the new media into the I/O slots.
2. Close the door.
   The library locks the door and moves the media into the storage area. You can import only 100 volumes at one time.
   a. If the library is configured with `access=private`, the library informs the daemon as the media is moved and the media is added to the catalog.
   b. If the library is configured with `access=shared`, issue the `samimport`(1M) command to add the media to the catalog.

How To Export a Cartridge

1. Use the `export`(1M) command to export cartridges.
   This command moves the media to the I/O area and turns on the output mode light on the operator panel.
2. Physically remove the media from the I/O area.

Sony Direct Attached 8400 PetaSite Automated Libraries

> Note -
> Sony 8400 PetaSite libraries are not supported by the SAM-QFS software on an x64 hardware platform.

The Sony 8400 PetaSite Series automated library is different from other Sony models because it has an eight-slot import and export mailbox (slots 400-407). Because the mailbox slots can be used as storage slots, the SAM-QFS library catalog keeps track of the mailbox slots. This automated library uses a barcode reader.

> Note -
> The information in this section applies only to Sony direct attached 8400 PetaSite automated libraries. This information does not pertain to the Sony direct attached B9 and B35 automated libraries, nor does it pertain to the Sony Network Attached Automated Libraries .

How To Import Tapes

1. Open the door of the automated library by pushing the open/close button on the front panel of the automated library.
2. Load the cartridges into the mailbox slots.
3. Push the open/close button on the front panel of the automated library
4. Manually close the door to the mailbox.
   The automated library checks the mailbox slots for the cartridge barcodes. If the library detects a problem with a barcode, both the `in` and `out` lights flash for that slot.
5. Use the `samimport`(1M) command to enable the SAM-QFS system to recognize the imported cartridges.

```
samimport <eq>
```

   For eq, specify the equipment ordinal of the device being addressed as defined in the `mcf` file.
   You can also perform this step by using SAM-QFS Manager. For more information, see SAM-QFS Manager online help.

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

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

```
move <source-slot> <destination-slot> <eq>
```

Argument	Meaning
source-slot	The number of the slot in which the cartridge currently resides.
destination-slot	The number of the slot into which the cartridge should be moved.
eq	The equipment ordinal of the device being addressed as defined in the `mcf` file.

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

```
samexport <eq>
```

For eq, specify the equipment ordinal of the device being addressed as defined in the `mcf` file.
You can also perform this step by using SAM-QFS Manager. For more information, see SAM-QFS Manager online help.

How To Export a Tape Using Mailbox Slots As Storage Slots

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

```
samexport <eq>
```

For eq, specify the equipment ordinal of the device being addressed as defined in the `mcf` file.
You can also perform this step by using SAM-QFS Manager. For more information, see SAM-QFS Manager online help.

How To Move a Cartridge to a Different Slot

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

```
move <eq>:<source-slot> <destination-slot>
```

Argument	Meaning
eq	The equipment ordinal of the device being addressed as defined in the `mcf` file.
source-slot	The number of the slot in which the cartridge currently resides.
destination-slot	The number of the slot into which the cartridge should be moved.

You can also perform this step by using SAM-QFS Manager. For more information, see SAM-QFS Manager online help.

Sony Network Attached Automated Libraries

> Note -
> Sony network attached libraries are not supported by the SAM-QFS software on an x64 hardware platform.

The following sections describe aspects of this library's operations that differ from the procedures described in Managing Automated Libraries and Manually Loaded Drives.

Because you use vendor-supplied utilities to physically add and remove cartridges, the SAM-QFS interfaces (`samimport`, `samexport`, and SAM-QFS Manager) affect only the library catalog.

How To Import a Cartridge

1. Use Sony commands to physically move the cartridge into the library.
2. Use the `samimport`(1M) command to update the library catalog.

```
samimport -v "<volser>" <eq>
```

Argument	Meaning
volser	The volser to be added. The PSC API interface verifies that the Sony automated library has the volser information before updating the library catalog with the new entry. If the cartridge does not physically exist in the library, the entry is placed in the historian catalog. The volser value must be enclosed in quotation marks if it contains spaces.

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

How To Export a Cartridge

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

```
samexport <eq>:<slot>
samexport <media-type>.<vsn>
```

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

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

StorageTek ACSLS-Attached Automated Libraries

The following sections describe aspects of this library's operations that differ from the procedures described in Managing Automated Libraries and Manually Loaded Drives.

Some StorageTek automated libraries, such as the StorageTek 9730 library use a mail slot to import and export only one cartridge at a time. A mailbox is an area used for putting cartridges into and removing cartridges from the automated library. Examples of StorageTek automated libraries that have a mailbox are the StorageTek 9714 and the StorageTek 9710 libraries.
In StorageTek documentation, the mailbox and mailslot are both referred to as the cartridge access port (CAP).

When importing and exporting cartridges from any ACSLS-attached automated library, you must keep the ACSLS inventory and the SAM-QFS catalog in agreement.

- When importing cartridges, the `samimport`(1M) command does not insert cartridges physically into the automated library. You must also issue ACSLS commands to complete the operation.
- When exporting cartridges, issue the `samexport`(1M) command with its `-f` option to direct the SAM-QFS system to put the cartridge in the CAP and to update the catalog. Without the `-f` option, the cartridge is not in the CAP so you must then use ACSLS commands to complete the operation.

You can also perform the import and export procedures by using `samu`(1M) or SAM-QFS Manager. For more information, see SAM-QFS Manager online help.

How To Import Tapes

- Use the `samimport`(1M) command in the following format:

```
samimport -v <vsn eq>
```

Argument	Meaning
vsn	The volume serial name (VSN) assigned to the volume.
eq	The equipment ordinal of the device being addressed as defined in the `mcf` file.

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

How To Export Tapes Using a Mailbox

You can export tape cartridges by slot or by VSN. Export the tape using one of the following formats:

- ```
  samexport [-f] <eq>:<slot>
  ```

- ```
  samexport [-f] <media-type>.<vsn>
  ```

Argument	Meaning
`-f`	Specification for the SAM-QFS system to put the volume in the cartridge access port (CAP) and to update the catalog accordingly.
eq	The equipment ordinal of the device being addressed as defined in the `mcf` file.
slot	The number of a storage slot in an automated library as recognized in the library catalog.
media-type	The media type. For a list of valid media types, see the `mcf`(4) man page.
vsn	The volume serial name (VSN) assigned to the volume.

About Archiving

Archiving is the process of copying a file from a file system to a volume that resides on a removable media cartridge or on a disk partition of another file system. When using the Sun Storage Archive Manager (SAM), you can specify that files be archived immediately, specify that files never be archived, and perform other tasks.

The term archive media is used to refer to the various cartridges or disk slices to which archive volumes are written. This page provides general guidelines for developing archive policies for your site and describes the archiver's theory of operations. See Configuring the Archiver for details on how to configure the archiver.

Planning Archiving Operations

The archiver automates storage management operations using the `archiver.cmd` file. Before writing this file, review the following general guidelines. They can improve the performance of your file system and the archiver, which can help ensure that your data is stored in the safest way possible.

- Save your archive logs. The archive logs are essential to recovering data, even if the SAM software is unavailable. Keep these logs in a safe place in the event of a catastrophic disaster.
- Use regular expressions for volume ranges. Enable the system to put files on many different volumes. Volume ranges allow the system to run continuously. If you use specific volume names, the archive set copies can fill a volume rapidly, causing workflow problems as you replace media.
- Set an optimal archive interval. The archive interval is the time between file system scans. Set your archive interval based on how often files are created and modified, and whether you want to save all modified copies. An archive interval that is too short keeps the archiver scanning almost continuously.
- Consider the number of file systems you are using. Multiple file systems increase the performance of the archiver over a single file system. Multiple file systems can be scanned in less time than a single file system.
- Use directory structures to organize your files in a file system. To maximize performance, do not place more than 10,000 files in a directory.
- Always make a minimum of two file copies on two separate volumes. Putting data on a single media type puts your data at risk if physical problems with the media occur.
- Back up your metadata (directory structure, file names, and so on) Use `samfsdump`(1M) on a regular basis. The metadata is stored in an archive set that has the same name as the file system. You can use this information to recover a file system in the event of a disaster. If you do not want the archiver to back up your metadata, you can set `archivemeta=off` in the `archiver.cmd` file and schedule the `samfsdump` command to run in a `cron` file.

Preview Queue

The archiver and stager processes both can request that media be loaded and unloaded. If the number of requests exceeds the number of drives available for media loads, the excess requests are sent to the preview queue.

By default, preview requests are satisfied in first-in-first-out (FIFO) order. You can override the FIFO default by entering directives in the preview command file, which is written to `/etc/opt/SUNWsamfs/preview.cmd`. For more information about this file and setting priorities for archiving and staging, see Prioritizing Preview Requests.

Archiver Daemons

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

When Sun Storage Archive Manager (SAM) is initiated, its `sam-fsd` daemon starts the `sam-archiverd` daemon.
The `sam-archiverd` daemon executes the `archiver`(1M) command to read the `archiver.cmd` file and builds the tables necessary to control archiving.

The `sam-archiverd` daemon starts a `sam-arfind` process for each mounted file system. When a file system is unmounted, it stops the associated `sam-arfind` process.

The `sam-archiverd` process monitors `sam-arfind` and processes signals from an operator or other processes.

Archiving Operations

Operator actions are not required to archive files. The archiver writes files to a volume on the archive media. Archive media can contain one or more volumes. Each volume is identified by a unique identifier called a volume serial name (VSN).

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

To be a candidate for archiving or re-archiving, the data in a file must change. A file is not archived if it is accessed but not changed. For example, issuing a `touch`(1) command on a file does not cause it to be archived or re-archived. If the `mv`(1) command is used to rename a file, the file might move to a different archive set. In this case, the archiver software re-evaluates the archive copies to determine whether the file needs to be archived or re-archived.

> Note
> Because issuing an `mv`(1) command alters the file name but not the file data, this action can have ramifications for disaster recovery if you plan to restore from `tar`(1) files. For more information about disaster recovery, see SAM-QFS Troubleshooting.

The archive operation is affected by the following factors:

- archive age The period of time that has passed since the file was last modified. The archive age can be defined for each archive copy. Use the `touch`(1) command to change the default time references on their files to values in the past or future. However, this practice can cause unexpected archiving results. To avoid problems, the archiver adjusts the references so that they are always at a point between the file creation time and the present time.
- archive priority This value is computed from file property characteristics and from file property multipliers associated with the archive set, as follows:

  ```
  <archive-priority> = <file-property-value> x <property-multiplier>
  ```

  - Most file-property-value numbers are 1 (for true) or 0 (for false). For instance, the value of the property copy 1 is 1 if archive copy 1 is being made. The values of copy 2, copy 3, and copy 4 are therefore 0. Other properties, such as archive age and file size, can have values other than 0 or 1.
  - The property-multiplier value is determined from the `-priority` parameters for the archive set. Various aspects of a file, such as age or size, can be given values to determine the archive request's priority. The default value for all property multipliers is 0.0. For more information about the `-priority` parameter, see the `archiver.cmd`(4) man page.

The archive-priority and the property-multiplier values are floating-point numbers.

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

- Step 1 Identifying Files to Archive
- Step 2 Composing Archive Requests
- Step 3 Scheduling Archive Requests
- Step 4 Archiving the Files in an Archive Request
- Sample Default Output from archiver(1M) -l Command

Step 1 Identifying Files to Archive

A separate sam-arfind process monitors each mounted file system to determine the files that need archiving. The file system notifies this sam-arfind process whenever a file is changed in a manner that affects its archival state, such as file modification, re-archiving, unarchiving, and renaming.
The sam-arfind process examines the file to determine which archive set the file belongs to and what action to take.

To determine a file's archive set,the sam-arfind process uses the following file properties descriptions:

- The directory path portion of the file's name and, optionally, the complete file name using a regular expression
- The user name of the file's owner
- The group name of the file's owner
- A minimum file size
- A maximum file size

To determine the archive action: the sam-arfind process performs the following evalutation:

- If the archive age of the file for one or more copies has been met or exceeded, the sam-arfind process adds the file to one or more archive requests for the archive set. An archive request is a collection of files that belong to the same archive set. The archive request resides in the /var/opt/SUNWsamfs/archiver/file_sys/ArchReq directory. The files in this directory are binary files. To display them, use the showqueue(1M) command. Separate archive requests are used for files not yet archived and for files being re-archived. This allows scheduling to be controlled independently for these two types of files.

- If the archive age of the file for one or more copies has not been met, the directory in which the file resides and the time at which the archive age is reached is added to a scan list. Directories are scanned as the scan list times are reached. Files that have reached their archive age are added to archive requests.

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

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

The archive action is accomplished using two methods:

- Continuous Archiving
- Scanned Archiving

Continuous Archiving

With the continuous archiving method, the archiver works with the file system to determine which files need to be archived.
Continuous archiving is the default archiving method (the archiver.cmd file parameter is examine=noscan) and operates with the following default start conditions:

- The archive starts every two hours.
- The archive waits until there is at least 90% of the archmax value of data to archive.
- The archive waits until there is at least 500,000 files to archive.
  When any of the scheduling start conditions is reached, the sam-arfind process sends each archive request to the archiver daemon, sam-archiverd, to set the schedule for file copying to archive media.

To control the schedule of the archive operation, specify the start conditions for each archive set by using the -startage, -startcount, and -startsize parameters. These conditions enable you to optimize archive timeliness versus archive work done. For example:

- If creating files that you want archived together takes an hour, set the -startage parameter to one hour (-startage 1h) to ensure that all files are created before the archive request occurs.
- If you want all archive requests to be no less than 150 gigabytes of data, specify a -startsize of 150 gigabytes (-startsize 150g) to direct the archiver to wait until 150 gigabytes of data are ready to be archived.
- If you know that 3000 files are available for archival, specify -startcount 3000 to ensure that the files are archived together.

For more information about archiving parameters, see Global Archiving Directives.

Scanned Archiving

With the scanning method, the archiver checks the file systems periodically and selects files for archiving. To enable the scanning method and to disable continuous scanning, specify examine=scan in the archiver.cmd file.

The sam-arfind process scans each file system periodically to determine which files need archiving and to place them in archive requests. The first scan is a directory scan, in which sam-arfind descends recursively through the directory tree. The process examines each file and sets the file status flag to archdone if the file does not need archiving. During successive scans, the sam-arfind process scans the .inodes file and examines only inodes without the archdone flag.

When the scan is complete, the sam-arfind process sends each archive request to the archiver daemon, sam-archiverd, to be scheduled for file copying to archive media. The sam-arfind process then sleeps for the duration specified by the interval=time directive. At the end of the interval, the sam-arfind process resumes scanning.

For information about controlling the setting of the archdone flag, see The setarchdone Directive: Controlling the Setting of the archdone Flag.

Step 2 Composing Archive Requests

Using either archiving methods, the result is that the sam-arfind process sends each archive request to the archiver daemon, sam-archiverd for composing. This section describes the composing process.

Composing is the process of selecting the files from the archive request to be archived at one time. Because of the capacity of the archive media or of the controls specified in the archiver command file, the files in an archive request might not be archived all at the same time. When the archive copy operation is complete for an archive request, the archive request is recomposed if files remain to be archived.

The sam-archiverd daemon places the files in the archive request according to certain default and site-specific criteria. The default operation is to archive all the files in an archive request by their full path name so that files in the same directories are also together on the archive media. The site-specific criteria enable you to control the order in which files are archived and how they can be distributed on volumes. These criteria, called archive set parameters, are evaluated in the following order: -reserve, -sort, -rsort (reverse sort), and -drives. For more information, see the archiver.cmd(4) man page.

When the -reserve owner parameter is specified, the sam-archiverd daemon orders the files in the archive request according to the file's directory path, user name, or group name. The files belonging to the owner are selected for archiving. The remaining files are archived later.

When the -sort or -rsort method parameter is specified, the sam-archiverd daemon orders the files according to the specified sort method such as age, size, or directory location.

When an archive request contains both online and offline files, the online files are selected for archiving first.
In the absence of a specified sort method, the offline files are ordered by the volume on which the archive copies reside. This rule ensures that all files in each archive set on the same volume are staged at the same time in the order in which they were stored on the media. When more than one archive copy of an offline file is being made, the offline file is not released until all required copies are made. All the files to be staged from the same volume as the first file are selected for archiving.

> Note
> The -sort and -rsort parameters can have a negative effect on performance during archiving of offline files if the order of the files to be archived does not match the order of the volumes needed for the offline files. Use these parameters only for the first archive copy to be made. Other copies must maintain the order of the first copy if enough archive media space is available when the copies are started.

After being composed, the archive requests are entered in the sam-archiverd daemon's scheduling queue, as described in the next section.

Step 3 Scheduling Archive Requests

The scheduler in the sam-archiverd daemon executes on demand when one of the following conditions exists:

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

The archive requests in the scheduling queue are ordered by priority. Each time the scheduler runs, it examines all archive requests to determine whether they can be assigned to a sam-arcopy process to have their files copied to archive media.

The following must be true in order for archive requests to be scheduled:

- Drives must be available for making file copies.
- Volumes must be available and have sufficient space to hold the files in the archive request.

Drives

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

If the total size of files in the archive request is less than the `-drivemin` value, one drive is used. The `-drivemin` value is either the value specified by the `-drivemin` parameter or the `archmax` value. The `archmax` value is specified by the `-archmax` parameter or the value defined for the media. For more information about the `-archmax` parameter and the `archmax=` directive, see the `archiver.cmd`(4) man page.

If the total size of files in the archive request is more than the `-drivemin` value, the number of drives used is determined by the total size of the files divided by the `-drivemin` value. If the number of drives available is fewer than that specified by the `-drives` parameter, the smaller number of drives is used.

Drives can take varying amounts of time to archive files. The `-drivemax` parameter specifies the maximum number of bytes to be written to a drive before that drive is rescheduled for more data. You can use the `-drivemax` parameter to obtain better drive utilization.

Volumes

For archiving to occur, at least one volume must have enough space to hold at least some of the files in the archive request. The volume that has most recently been used for the archive set is the one scheduled, if it has enough space. This volume must not be in use by the archiver.

If a volume usable for the archive set is busy, another is selected, unless the `-fillvsns` parameter is specified. In this case, the archive request cannot be scheduled.

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

You can specify volume overflow for the archive set by using the `-ovflmin` parameter, or for the media by using the `ovflmin=` directive. For more information about the `-ovflmin` parameter and the `ovflmin=` directive, see the `archiver.cmd`(4) man page. The `ovflmin` specification determines the file size threshold above which additional volumes or media are assigned for archiving. An `ovflmin` value specified for the archive set takes precedence over an `ovflmin` value specified for the media.

If the size of the files is less than the value of `ovflmin`, the files cannot be archived. An appropriate message for this condition is sent to the log. If the size of the files is more than the value of `ovflmin`, additional volumes are assigned as required. Volumes are selected in order of decreasing size in order to minimize the number of volumes required. If no usable volumes can be found for the archive request, the archive request waits until volumes become available.

Certain properties, such as whether the file is online or offline, are used in conjunction with the archive priority to determine the scheduling priority for a particular archive request. For more information about customizing the priority multiplier, see the `-priority` parameters described on the `archiver.cmd`(4) man page.

For each archive request, the `sam-archiverd` daemon computes the scheduling priority by adding the archive priority to multipliers associated with various system resource properties. These properties are associated with the number of seconds for which the archive request has been queued, whether the first volume to be used in the archiving process is loaded into a drive, and so on.

Using the adjusted priorities, the `sam-archiverd` daemon assigns each ready archive request to be copied, as described in the next section.

Step 4 Archiving the Files in an Archive Request

When an archive request is ready to be archived, the `sam-archiverd` daemon marks the archive file boundaries so that each archive file's size is less than the specified `-archmax` value. If a single file is larger than this value, it becomes the only file in an archive file.

For each archive request and each drive to be used, the `sam-archiverd` daemon assigns the archive request to a `sam-arcopy` process to copy the files to the archive media. The archive information is entered into the inode.

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

For each file that was staged, the disk space is released until all files in the list have been archived.

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

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

You can also specify timeouts for archiving operations in the `archiver.cmd` file. The directive is as follows:

```
timeout = [<operation> | <media>] <time>
```

operation is one of the following:

- `read`- Reading the file from the disk. The default timeout is 1 minute.
- `request`- Requesting the archive media. The default timeout is 15 minutes.
- `stage`- Staging the file to be archived. The default timeout is 0 minutes.
- `write`- Writing to the archive media. The default timeout is 15 minutes.

The timeout value for the write operation can also be specified for individual media.

Notification of Schedule Queue Time

You can configure the `-queue_time_limit` time archive set parameter in the `archiver.cmd` file to notify the administrator when an archive request has been in the schedule queue longer than a certain amount of time. If the archive request remains in the queue at the end of this time, an email is sent to the system administrator.

Sample Default Output from archiver(1M) -l Command

The following shows sample output from the `archiver(1M) -l` command.

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

all path:
copy:1 arch_age:30
Archive sets:
all
copy:1 media:mo
big
copy:1 media:lt
samfs1
copy:1 media:mo
```

Log Files and Event Logging for Archive Operations

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 `sam-arfind` and `sam-arcopy` processes use the `syslog` facility and `archiver.sh` to log warnings and informational messages in a log file that contains information about each archived or automatically unarchived file.

The log file is disabled by default. Use the `logfile=` directive in the `archiver.cmd` file to enable logging and to specify the name of the log file. For more information about the log file, see About the archiver.cmd File and the `archiver.cmd`(4) man page.

The following example shows sample lines from an archiver log file. The following table defines each field in the log.

Example - Archiver Log File Lines

```
A 2001/03/23 18:42:06 mo 0004A arset0.1 9a089.1329 samfs1 118.51 162514 t0/fdn f 0 56
A 2001/03/23 18:42:10 mo 0004A arset0.1 9aac2.1 samfs1 189.53 1515016 t0/fae f 0 56
A 2001/03/23 18:42:10 mo 0004A arset0.1 9aac2.b92 samfs1 125.53 867101 t0/fai f 0 56
A 2001/03/23 19:13:09 lt SLOT22 arset0.2 798.1 samfs1 71531.14 1841087 t0/fhh f 0 51
A 2001/03/23 19:13:10 lt SLOT22 arset0.2 798.e0e samfs1 71532.12 543390 t0/fhg f 0 51
A 2003/10/23 13:30:24 dk DISK01/d8/d16/f216 arset4.1 810d8.1 qfs2 119571.301 1136048 t1/fileem f 0
0
A 2003/10/23 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.8ad qfs2 119573.295 1849474 t1/fileud f
0 0
A 2003/10/23 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.16cb qfs2 119576.301 644930 t1/fileen f
0 0
A 2003/10/23 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.1bb8 qfs2 119577.301 1322899 t1/fileeo
f 0 0
```

Table - Archiver Log File Fields

Field	Example Value	Content
1	A	Archive activity, as follows: • A for archived. • R for rearchived. • U for unarchived.
2	2001/03/23	Date of the archive action, in yyyy/mm/dd format.
3	18:42:06	Time of the archive activity, in hh:mm:ss format.
4	mo	Archive media type. For information about media types, see the mcf(4) man page.
5	0004A	VSN. For removable media cartridges, the volume serial name. For disk archives, the disk volume name and archive tar (1) file path.
6	arset0.1	Archive set and copy number.
7	9a089.1329	Physical position of the start of the archive file on media (tar(1) file) and file offset within the archive file, in hexadecimal format.
8	samfs1	File system name.
9	118.51	Inode number and generation number. The generation number is used in addition to the inode number for uniqueness because inode numbers are reused.
10	162514	Length of the file if the file is written on only one volume. Length of the section if the file is written on multiple volumes.
11	t0/fdn	Path and name of the file relative to the file system's mount point.
12	f	Type of file, as follows: • d for directory • f for regular file • l for symbolic link. • R for removable media file • I for segment index • S for data segment
13	0	Section of an overflowed file or segment. If the file is an overflowed file, the value is nonzero. For all other file types, the value is 0.
14	56	Equipment ordinal of the drive on which the file was archived.

Data Verification

You can enable data verification for archive copies. This feature checks for data corruption on any data that is copied to secondary and/or tertiary media.

The data verification process performs a read-after-write verification test and records a confirmation of data validity in the metadata properties for that file. The process uses the ssum option to mark files and directories to be verified. The normal checksum method is employed to verify copies written to tape or disk archive.

Use the ssum -e command to set data verification for a file or directory. Child directories inherit the data verification properties of their parent. This command forces the generation and use of checksums for archiving and staging, and prevents the release of the file until all archive copies have been created and their checksums verified. Only a superuser can set this attribute on a file or directory.

> Note
> Data verification places an additional burden on stager resources because data verification requests are placed on the stager queue in addition to normal stage requests. Data verification also leads to additional tape mounts and therefore affects archiver and stager performance. Because a file cannot be released until all archive copies are made, using data verification might also require additional disk cache.

Configuring the Archiver

This page shows how to configure the archiver by modifying the archiver.cmd file.

About the **archiver.cmd** File

The archiver.cmd file controls the archiver's behavior.

The archiver copies files from a file system to volumes on removable media cartridges or to disk partitions in another file system. You can tune the archiver operations to suit the types of files at your site and to suit your site's data protection needs by creating an archiver command file named /etc/opt/SUNWsamfs/archiver.cmd. You are not required to create an archiver.cmd file, but the efficiency and performance of the archiver is improved if you tune the archiver to your site.

By default, the archiver starts whenever the sam-fsd process is started and a file system is mounted.
If the archiver.cmd file does not exist, the archiver is put in a wait state. To restart the archiver, use the samcmd arrun command. If no archiver.cmd file is found after the restart, the archiver continues to run, using the following default settings:

- Archives all files to all available volumes in all configured libraries.
- Makes one copy of each file.
- Sets the archive age for all files to 4 minutes. The archive age is the amount of time since a file's last modification.
- Sets the archive interval to 10 minutes. The archive interval is the amount of time that elapses between complete archiving processes.

To tune the actions of the archiver for your site, set directives in the archiver.cmd. A directive acts like a command parameter and consist of lines of text in the archiver.cmd file. The following rules apply to the archiver.cmd file:

- Each directive line contains one or more fields separated by spaces or tabs.
- Any text that appears after the pound sign character (#) is treated as a comment and is not examined.
- Lines that end with a backslash character () are joined to the next line.
  The archiver.cmd file uses two types of directives: Archive Directives and Archive Set Directives.
  For comprehensive information about the archiver.cmd directives, see the archiver.cmd(4) man page.

Whenever you make changes to the archiver.cmd file, check for syntax errors using the archiver(1M) command. This command produces a list of all options and writes a list of the volumes, file system content, and errors to the standard output file (stdout). If you encounter errors, correct them in the file and run the archiver(1M) command again to verify your corrections. The archiver does not archive any files if it finds errors in the archiver.cmd file.

Archive Directives

Archive directives specify the general archive operations and consist of two main areas in the archiver.cmd file:

- At the top of the file, global directives affect all file systems defined in your mcf file.
- The lower part of the file contains file-system-specific directives. These directives must come after the global directives. For any file system, these directives override the global directives. The file-system-specific directives start with an fs=name directive that identifies

the file system.
Within a file-system's directives, you can also specify archive copy directives to customize the number and frequency of the archive copies.

See the Archive Directives page for detailed information.

Archive Set Directives

An archive set identifies a group of files to be archived, regardless of their file system. A file in a file system can be a member of only one archive set. Files in an archive set share common criteria that pertain to the size, ownership, group, or directory location.
The archive set controls the destination of the archive copy, how long the copy is kept archived, and how long the software waits before archiving the data. All files in an archive set are copied to the volumes associated with that archive set.
The directives for an archive set include:

- Assignment directives define archive sets.
- Copy parameters define how each archive set is archived.
- VSN association directives assign volumes to archive sets.
- VSN pools directives define a collection of volumes.

See the Archive Set Directives page for detailed information.

How to Start the Archiver

The archiver starts automatically when a file system is mounted.
If the `archiver.cmd` file does not exist, the archiver is put in a `wait` state.

To restart the archiver, use the `samcmd arrun` command.
If the `archiver.cmd` file does not exist, the archiver runs in its default manner, as described in About the archiver.cmd File.

How to Create an **`archiver.cmd`** File Using the Command Line

- If your site has an `/etc/opt/SUNWsamfs/archiver.cmd` file and your system is already archiving files, do not make changes to it. Copy the file to a location where you can edit and test it. When it is verified, replace the existing file with the new one.
- If your site does not have a `archiver.cmd` file, you can edit the file in the `/etc/opt/SUNWsamfs` directory.

1. Edit the `archiver.cmd` file to add or change the directives that control archiving at your site. For information about the directives you can include in this file, see Archive Directives and Archive Set Directives for details.
2. Save and close the `archiver.cmd` file.
3. Verify the file for the current SAM-QFS environment. If you are working with a test `archiver.cmd` file, use the `-c` option with the `archiver`(1M) command and supply the file name.

```
# archiver -lv
```

4. If you are working with a test file, move it to `/etc/opt/SUNWsamfs/archiver.cmd`.
5. Use the `samd config` command to propagate the file changes and restart the system.

```
# samd config
```

How to Create an **`archiver.cmd`** File Using SAM-QFS Manager

When you create or edit an archive policy for a file system within the SAM-QFS Manager interface, the `archiver.cmd` file is automatically created or edited.

1. On the Servers page, select the name of the server for which you want to create a policy.
The File Systems Summary page is displayed.
2. Select the Archive Administration node in the navigation tree.
The Archive Policies Summary page is displayed.
3. Click New.
The New Archive Policy wizard is displayed.
4. Follow the steps in the wizard.

For detailed instructions on using the New Archive Policy wizard, see the SAM-QFS Manager online help.
When you save the new archive policy, it is automatically written to the `archiver.cmd` file.

Example archiver.cmd Files

This section provides some examples of archiving configurations.

Example - Simple archiver.cmd File

The following example shows a simple `archiver.cmd` file that you can modify. Add directives only to accommodate more archive sets, copies, and VSN usage.

```
# archiver.cmd
# One file system = samfs1

archmax = sg 2G
examine = noscan

fs = samfs1
logfile = /var/opt/SUNWsamfs/log/samfs1.log
all_archset .
  1 -norelease 10m
  2 -norelease 10m

params
allsets -sort path -offline_copy stageahead -reserve set
allsets -recycle_hwm 50 -recycle_mingain 90 -recycle_vsncount 5 -recycle_dataquantity 40G
allsets.1 -startage 6h -startsize 6G -startcount 30000
allsets.2 -startage 10h -startsize 12G -startcount 60000 -archmax 12G
endparams

vsns
all.1 li .*
all.2 li .*
endvsns
```

Example - Advanced archiver.cmd File

The following example shows a complex `archiver.cmd` file. The comments indicate the various types of directives.

```
# Global directives

archmax = li 8G
examine = noscan
scanlist_squash = on

# limit the drives

drives = stk50 3

# File selection

fs = samfs1
logfile = /var/adm/samfs1.log
archive_archset .
  1 -norelease 5m
  2 -norelease 5m

fs = samfs2
logfile = /var/adm/samfs2.log
archive_archset .
  1 -norelease 5m
  2 -norelease 5m

fs = samfs3
logfile = /var/adm/samfs3.log
archive_archset .
  1 -norelease 5m
  2 -norelease 5m

fs = samfs4
logfile = /var/adm/samfs4.log
archive_archset .
  1 -norelease 5m
  2 -norelease 5m

fs = samfs5
logfile = /var/adm/samfs5.log
archive_archset .
  1 -norelease 5m
  2 -norelease 5m

# The below information is for tape archiving.
# The recycler is not set up to actually recycle. It is set up for
# data checks and file recovery.

params allsets -sort path -offline_copy stageahead -reserve set
allsets -recycle_ignore allsets -recycle_hwm 50 -recycle_mingain 90 -recycle_vsncount 1
allsets.1 -startage 6h -startsize 8G -startcount 90000 -drives 3 -drivemin 10G
allsets.2 -startage 10h -startsize 12G -startcount 90000 -archmax 12G -drives 3 -drivemin 10G
endparams

# Define VSNS for archive sets

vsns
archive.1 li .*
archive.2 li .*
endvsns
```

Example - No archiver.cmd File

This example illustrates the action of the archiver when no `archiver.cmd` file is used in a SAM-QFS environment that has one file system, an optical automated library with two drives, and six cartridges.

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

```
# archiver -lv
Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh
Archive media:
media:lt archmax: 512.0M Volume overflow not selected
media:mo archmax: 4.8M Volume overflow not selected
```

The following `archiver -lv` output indicates that the archiver uses two drives. It lists the 12 volumes, storage capacity, and VSNs with available space.

```
Archive libraries:
Device:hp30 drives_available:2 archive_drives:2
Catalog:
mo.optic00 capacity: 1.2G space: 939.7M -il-o-------
mo.optic01 capacity: 1.2G space: 934.2M -il-o-------
mo.optic02 capacity: 1.2G space: 781.7M -il-o-------
mo.optic03 capacity: 1.2G space: 1.1G -il-o-------
mo.optic10 capacity: 1.2G space: 85.5M -il-o-------
mo.optic11 capacity: 1.2G space: 0 -il-o-------
mo.optic12 capacity: 1.2G space: 618.9k -il-o-------
mo.optic13 capacity: 1.2G space: 981.3M -il-o-------
mo.optic20 capacity: 1.2G space: 1.1G -il-o-------
mo.optic21 capacity: 1.2G space: 1.1G -il-o-------
mo.optic22 capacity: 1.2G space: 244.9k -il-o-------
mo.optic23 capacity: 1.2G space: 1.1G -il-o-------
```

The following `archiver -lv` output shows that the archive set `samfs` includes both metadata and data files. The archiver makes one copy of the files when their archive age reaches the default four minutes (240 seconds).

```
Archive file selections:
Filesystem samfs Logfile:
samfs Metadata
copy:1 arch_age:240
samfs1 path:.
copy:1 arch_age:240
```

The following `archiver -lv` output shows the files in the archive sets archived to the volumes in the indicated order.

```
Archive sets:
allsets
samfs.1
media: mo (by default)
Volumes:
optic00
optic01
optic02
optic03
optic10
optic12
optic13
optic20
optic21
optic22
optic23
Total space available: 8.1G
```

Example - Data Files Archived Separately From Metadata

This example shows how to separate data files into two archive sets separate from the metadata. The environment includes a manually mounted DLT tape drive in addition to an optical automated library. The big files are archived to tape, and the small files are archived to optical cartridges.

The following example shows the content of the `archiver.cmd` file.

```
# archiver -lv -c example2.cmd
Reading archiver command file "example2.cmd"
1: # Example 2 archiver command file
2: # Simple selections based on size
3:
4: logfile = /var/opt/SUNWsamfs/archiver/log
5: interval = 5m
6:
7: # File selections.
8: big . -minsize 500k
9: all .
10: 1 30s
11:
12: vsns
13: samfs.1 mo .*0[0-2] # Metadata to optic00 - optic02
14: all.1 mo .*0[3-9] .*[1-2][0-9] # All others for files
15: big.1 lt .*
16: endvsns
```

The following archiver -lv output shows the media and drives to be used.

```
Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh
Archive media:
media:lt archmax: 512.0M Volume overflow not selected
media:mo archmax: 4.8M Volume overflow not selected
Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
Catalog:
mo.optic00 capacity: 1.2G space: 939.7M -il-o-------
mo.optic01 capacity: 1.2G space: 934.2M -il-o-------
mo.optic02 capacity: 1.2G space: 781.7M -il-o-------

mo.optic03 capacity: 1.2G space: 1.1G -il-o-------
mo.optic04 capacity: 1.2G space: 983.2M -il-o-------
mo.optic10 capacity: 1.2G space: 85.5M -il-o-------
mo.optic11 capacity: 1.2G space: 0 -il-o-------
mo.optic12 capacity: 1.2G space: 618.9k -il-o-------
mo.optic13 capacity: 1.2G space: 981.3M -il-o-------
mo.optic20 capacity: 1.2G space: 1.1G -il-o-------
mo.optic21 capacity: 1.2G space: 1.1G -il-o-------
mo.optic22 capacity: 1.2G space: 244.9k -il-o-------
mo.optic23 capacity: 1.2G space: 1.1G -il-o-------
Device:lt40 drives_available:0 archive_drives:0
Catalog:
lt.TAPE01 capacity: 9.5G space: 8.5G -il-o-------
lt.TAPE02 capacity: 9.5G space: 6.2G -il-o-------
lt.TAPE03 capacity: 9.5G space: 3.6G -il-o-------
lt.TAPE04 capacity: 9.5G space: 8.5G -il-o-------
lt.TAPE05 capacity: 9.5G space: 8.5G -il-o-------
lt.TAPE06 capacity: 9.5G space: 7.4G -il-o-------
```

> Note -
> The archiver(1M) -lv command shows only VSNs with space available.

The following archiver -lv output shows the organization of the file system. Files bigger than 512000 bytes (500 kilobytes) are archived after four minutes. All other files are archived after 30 seconds.

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

The following `archiver -lv` output shows the division of the archive sets among the removable media.

```
Archive sets:
allsets
all.1
media: mo
Volumes:
optic03
optic04
optic10
optic12
optic13
optic20
optic21
optic22
optic23
Total space available: 6.3G
big.1
media: lt
Volumes:
TAPE01
TAPE02
TAPE03
TAPE04
TAPE05
TAPE06
Total space available: 42.8G
samfs.1
media: mo
Volumes:
optic00
optic01
optic02
Total space available: 2.6G
```

Example - User and Data Files Archived to Various Media

In this example, user files and project data files are archived to various media. Files from the directory `data` are segregated by size to optical and tape media. Files assigned to the group ID `pict` are assigned to another set of volumes. Files in the directories `tmp` and `users/bob` are not archived. Archiving is performed at 15-minute intervals, and an archiving record is kept.

```
# archiver -lv -c example3.cmd
Reading archiver command file "example3.cmd"
1: # Example 3 archiver command file
2: # Segregation of users and data
3:
4: interval = 30s
5: logfile = /var/opt/SUNWsamfs/archiver/log
6:
7: no_archive tmp
8:
9: fs = samfs
10: no_archive users/bob
11: prod_big data -minsize 50k
12: 1 1m 30d
13: 2 3m
```

```
14: prod data
15: 1 1m
16: proj_1 projs/proj_1
17: 1 1m
18: 2 1m
19: joe . -user joe
20: 1 1m
21: 2 1m
22: pict . -group pict
23: 1 1m
24: 2 1m
25:
26: params
27: prod_big.1 -drives 2
28: prod_big.2 -drives 2
29: endparams
30:
31: vsns
32: samfs.1 mo optic0[0-1]$
33: joe.1 mo optic01$
34: pict.1 mo optic02$
35: pict.2 mo optic03$
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/opt/SUNWsamfs/scripts/archiver.sh
Archive media:
media:lt archmax: 512.0M Volume overflow not selected
media:mo archmax: 4.8M Volume overflow not selected
Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
Catalog:
mo.optic00 capacity: 1.2G space: 939.7M -il-o-------
mo.optic01 capacity: 1.2G space: 934.2M -il-o-------
mo.optic02 capacity: 1.2G space: 781.7M -il-o-------
mo.optic03 capacity: 1.2G space: 1.1G -il-o-------
mo.optic04 capacity: 1.2G space: 983.2M -il-o-------
mo.optic10 capacity: 1.2G space: 85.5M -il-o-------
mo.optic11 capacity: 1.2G space: 0 -il-o-------
mo.optic12 capacity: 1.2G space: 618.9k -il-o-------
mo.optic13 capacity: 1.2G space: 981.3M -il-o-------
mo.optic20 capacity: 1.2G space: 1.1G -il-o-------
mo.optic21 capacity: 1.2G space: 1.1G -il-o-------
mo.optic22 capacity: 1.2G space: 244.9k -il-o-------
mo.optic23 capacity: 1.2G space: 1.1G -il-o-------
Device:lt40 drives_available:0 archive_drives:0
Catalog:
lt.TAPE01 capacity: 9.5G space: 8.5G -il-o-------
lt.TAPE02 capacity: 9.5G space: 6.2G -il-o-------
lt.TAPE03 capacity: 9.5G space: 3.6G -il-o-------
lt.TAPE04 capacity: 9.5G space: 8.5G -il-o-------
lt.TAPE05 capacity: 9.5G space: 8.5G -il-o-------
lt.TAPE06 capacity: 9.5G space: 7.4G -il-o-------
Archive file selections:
Filesystem samfs Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
copy:1 arch_age:240
no_archive Noarchive path:users/bob
prod_big path:data minsize:50.2k
copy:1 arch_age:60 unarch_age:2592000
copy:2 arch_age:180
prod path:data
copy:1 arch_age:60
proj_1 path:projs/proj_1
copy:1 arch_age:60
copy:2 arch_age:60
joe path:. uid:10006
copy:1 arch_age:60
copy:2 arch_age:60
pict path:. gid:8005
copy:1 arch_age:60
```

```
copy:2 arch_age:60
no_archive Noarchive path:tmp
samfs path:.
copy:1 arch_age:240
Archive sets:
allsets
joe.1
media: mo
Volumes:
optic01
Total space available: 934.2M
joe.2
media: lt
Volumes:
TAPE01
TAPE02
Total space available: 14.7G
pict.1
media: mo
Volumes:
optic02
Total space available: 781.7M
pict.2
media: mo
Volumes:
optic03
Total space available: 1.1G
prod.1
media: mo
Volumes:
optic20
optic21
optic22
optic23
Total space available: 3.3G
prod_big.1
media: lt drives:2
Volumes:
TAPE03
TAPE04
Total space available: 12.1G
prod_big.2
media: lt drives:2
Volumes:
TAPE05
TAPE06
Total space available: 16.0G
proj_1.1
media: mo
Volumes:
optic10
Total space available: 85.5M
proj_1.2
media: mo
Volumes:
optic12
optic13
Total space available: 981.9M
samfs.1
media: mo
Volumes:
optic00
```

```
optic01
Total space available: 1.8G
```

Example - User and Data Files Archived to Optical Media

In this example, user files and project data files are archived to optical media.

Four VSN pools are defined. Three pools are used for user, data, and project, and one is a scratch pool. When `proj_pool` runs out of media, it relies on `scratch_pool` to reserve volumes. This example shows how to reserve volumes for each archive set based on the set component, owner component, and file system component. Archiving is performed at 10-minute intervals, and an archiving log is kept.

The following example shows the `archiver.cmd` file and archiver output.

```
Reading archiver command file "example4.cmd"
1: # Example 4 archiver command file
2: # Using 4 VSN pools
3:
4: interval = 30s
5: logfile = /var/opt/SUNWsamfs/archiver/log
6:
7: fs = samfs
8: users users
9: 1 10m
10:
11: data data
12: 1 10m
13:
14: proj projects
15: 1 10m
16:
17: params
18: users.1 -reserve user
19: data.1 -reserve group
20: proj.1 -reserve dir -reserve fs
21: endparams
22:
23: vsnpools
24: users_pool mo optic0[1-3]$
25: data_pool mo optic1[0-1]$
26: proj_pool mo optic1[2-3]$
27: scratch_pool mo optic2.$
28: endvsnpools
29:
30: vsn
31: samfs.1 mo optic00
32: users.1 mo -pool users_pool -pool scratch_pool
33: data.1 mo -pool data_pool -pool scratch_pool
34: proj.1 mo -pool proj_pool -pool scratch_pool
35: endvsns
Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh
Archive media:
media:mo archmax: 4.8M Volume overflow not selected
Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
Catalog:
mo.optic00 capacity: 1.2G space: 939.7M -il-o-------
mo.optic01 capacity: 1.2G space: 934.2M -il-o-------
mo.optic02 capacity: 1.2G space: 781.7M -il-o-------
mo.optic03 capacity: 1.2G space: 1.1G -il-o-------
mo.optic04 capacity: 1.2G space: 983.2M -il-o-------
mo.optic10 capacity: 1.2G space: 85.5M -il-o-------
mo.optic11 capacity: 1.2G space: 0 -il-o-------
mo.optic12 capacity: 1.2G space: 618.9k -il-o-------
mo.optic13 capacity: 1.2G space: 981.3M -il-o-------
mo.optic20 capacity: 1.2G space: 1.1G -il-o-------
mo.optic21 capacity: 1.2G space: 1.1G -il-o-------
mo.optic22 capacity: 1.2G space: 244.9k -il-o-------
mo.optic23 capacity: 1.2G space: 1.1G -il-o-------
```

```
Archive file selections:
Filesystem samfs Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
copy:1 arch_age:240
users path:users
copy:1 arch_age:600
data path:data
copy:1 arch_age:600
proj path:projects
copy:1 arch_age:600
samfs path:.
copy:1 arch_age:240
VSN pools:
data_pool media: mo Volumes:
optic10
Total space available: 85.5M
proj_pool media: mo Volumes:
optic12
optic13
Total space available: 981.9M
scratch_pool media: mo Volumes:
optic20
optic21
optic22
optic23
Total space available: 3.3G
users_pool media: mo Volumes:
optic01
optic02
optic03
Total space available: 2.7G
Archive sets:
allsets
data.1
reserve:/group/
media: mo
Volumes:
optic10
optic20
optic21
optic22
optic23
Total space available: 3.4G
proj.1
reserve:/dir/fs
media: mo
Volumes:
optic12
optic13
optic20
optic21
optic22
optic23
Total space available: 4.2G
samfs.1
media: mo
Volumes:
optic00
Total space available: 939.7M
users.1
reserve:/user/
media: mo
Volumes:
optic01
optic02
optic03
optic20
optic21
optic22
```

```
optic23
Total space available: 6.0G
```

About Disk Archiving

Archiving is the process of copying a file from online disk to archive media. With disk archiving, the archive medium is online disks in a file system.

Disk archiving can be implemented so that the files from one file system are archived to another file system on the same host or to another file system on a different host. When disk archiving is configured for two host systems, the systems act as a client and a server. The host system where the source files reside is the client and the host system where the archive copies are written is the server.

The file system to which the archive files are written can be any UNIX file system. However, if disk archive copies are written to a different host, that host must have at least one QFS or SAM-QFS file system installed on it.

The archiver treats files archived to disk volumes in the same way as files archived to volumes in a library. You can make one, two, three, or four archive copies. If you are making multiple archive copies, you can write one of the archive copies to disk volumes and write the other archive copies to removable media volumes. In addition, if you archive to disk volumes that are in an archived file system, the archive copies are also archived according to the `archiver.cmd` file rules for that file system.

When you plan disk archiving for your site, consider the following guidelines:

- Create disk volumes on a different disk from the one on which the original files reside.
- Make more than one archive copy and write to more than one type of archive media. For example, create archive copy 1 to disk volumes, copy 2 to tape, and copy 3 to magneto-optical disk.
- If you are archiving files to a file system on a server system, the archive copies can also be written to removable media cartridges in a library attached to that server system.

The following list summarizes some of the similarities and differences between archiving to disk volumes and archiving to removable media:

- Archive copies in disk volumes are not recorded in a catalog.
- Archive copies in disk volumes do not appear in the historian.
- Archiving to disk volumes does not rely on entries in the `mcf` file. Instead, you specify disk archive sets in the `archiver.cmd` file and you define disk volumes in `/etc/opt/SUNWsamfs/diskvols.conf`.
- To archive to disk volumes, you must define disk archive sets in the `archiver.cmd` file before mounting the file system. When you archive to removable media volumes, you can begin archiving after the file system is mounted without changing any of the default values in the `archiver.cmd` file.

About the diskvols.conf File

Disk archiving does not rely on entries in the `mcf` file. You specify disk archive sets in the `archiver.cmd` file and you define disk volumes in `/etc/opt/SUNWsamfs/diskvols.conf`.

Create the `diskvols.conf` file on the system where the source files reside. Depending on where the archive copies are written, the `diskvols.conf` file contains the following information:

- If the archive copies are written to a file system on the same host system, the `diskvols.conf` file defines the VSNs and the paths to each VSN.
- If the archive copies are written to a different host system, the `diskvols.conf` file contains the host name of that server system. In this case, create another `diskvols.conf` file on the server system that specifies the host names of the client systems.

> Caution
> Be very careful when configuring the recycler if you are using disk archiving in an environment with multiple SAM-QFS servers. The `diskvols.conf` file for each SAM-QFS server must point to a unique set of disk volume resource specifications (disk archiving target directories). If any of the target directories are shared by the SAM-QFS servers, running the recycler from one SAM-QFS server destroys the disk archive data that is managed by the other SAM-QFS server.

The `diskvols.conf` file must contain the following information:

```
<VSN_name> [<host_name>:] <path>
```

Field Name	Content
VSN_name	A unique alphanumeric name of up to 31 characters for the disk VSN to receive the archive copies.
host_name	The name of the host, followed by a colon character (:), to which archive copies are written. If you are archiving to a disk on another host, specify the name of the destination server. If you are archiving to a file system that resides on the same server as the source file system, do not specify the host name.
path	The full path, relative to the mount point, to the directory that receives the archive files. This directory must be in place before archiving can start, and the destination file system must be mounted. For example, if archive copies are written to the vsns directory in the archivefs1 file system, specify /archivefs1/vsns in the path field. Create the destination directory with write permission granted only to root.

The following additional rules apply to the diskvols.conf file:

- Start each comment line with a pound character (#). All text to the right of the # is ignored.
- To continue a line, put an apostrophe character (') at the end.

For more information, see the diskvols.conf(4) man page.

Disk Archiving Directives

When archiving to disk volumes, the archiver recognizes the directives that define archive sets and recycling and ignores directives that pertain to removable media cartridges. The archiver recognizes the directives in the following sections:

- Archive Set Copy Parameters
- Archive Directives
- Recycling Directives
- vsn Directives
- clients and endclients Directives
- -recycle minobs Recycler Directive

Archive Set Copy Parameters

All the parameters described in Archive Set Copy Parameters are valid except for the following:

- -reserve method
- -tapenonstop

To configure an archive set to write multiple, simultaneous disk archive streams, use the -drives parameter. In this configuration, volumes are selected in a round robin manner starting with the volume that has the highest percentage of available space. However, if the parameter '-fillvsns' is specified, the volume with the least percentage
remaining space is selected first.

Archive Directives

All the directives described in Archive Directives are valid except for the following:

- ovflmin min-size

Recycling Directives

All of the directives described in recycling directives are valid except for the following:

- -recycle_dataquantity size
- -recycle_vsncount count
- recycle_hwm

vsn Directives

The following directives are valid:

- vsns and endvsns
- vsnpools and endvsnpools

Disk volumes are supported in the VSN associations section and are defined with a dk media type. The volumes are identified by one or more

VSN expression keywords. You can also specify VSN pools from which disk volumes are to be selected as shown in the following example.

```
vsnpools
data_pool dk disk0[0-5]
endvsnpools

vsns
arset0.1 dk disk10 disk1[2-5]
arset1.1 dk -pool data_pool
endvsns
```

Disk archiving can be carried out on the Sun StorageTek 5800 system. The Sun StorageTek 5800 is an online storage appliance with an integrated hardware and software architecture in which the disk-based storage nodes are arranged in a symmetric cluster. The media abbreviation for Sun StorageTek 5800 disk archives in the `vsns` directives is `cb`.

> Note -
> If you are using the disk volumes on the Sun StorageTek 5800 for archiving, be aware that the Sun StorageTek 5800 is not a traditional file system and the security considerations are different from other types of disk storage. Read the Sun StorageTek 5800 documentation for more information.

clients and endclients Directives

The `clients` and `endclients` directives are valid. If you archive files from a client host to a server host, the server system must have a `diskvols.conf` file that contains the name of the client system. The format for these directives is shown in following example. For client-system, specify the host name of the client system that contains the source files.

```
clients
_client-system1_

_client-system2_
...
endclients
```

-recycle minobs Recycler Directive

The `-recycle_minobs` *`percent`* recycler directive is valid. This option is used to set a threshold for the recycler's rearchiving process for disk archives. The default threshold is 50 percent. When the percentage of obsolete files within an archived tar file on the disk reaches this threshold, the recycler moves the valid files from the archive into a new tar file. When all of the valid files have been moved, the original tar file is marked as a candidate for removal from the disk archive. This option is ignored for removable media recycling.

How to Enable Disk Archiving on the Host That Contains the Files to Be Archived

Perform this procedure on the host system that contains the files to be archived. As an alternative, you can use the SAM-QFS Manager interface to specify an archive policy that archives to disk volumes. This action updates both the `archiver.cmd` file and the `diskvols.conf` file.

> Note -
> If you are configuring a Sun QFS file system for the first time at your site and have therefore not yet installed the SAM-QFS software on another host, you must write the archive copies to disk volumes in a file system that is on the same host as the source files. If you configure a QFS file system on another host at a later time, you can modify your configuration files accordingly.

1. Become superuser on the host system that contains the files you want to archive.
2. Create or open the file `/etc/opt/SUNWsamfs/archiver.cmd`.
3. Add disk archive set directives as in the following example:

```
#
vsns
archset1.1 dk disk01
archset2.1 dk disk02
archset3.1 dk disk03
endvsns
```

Disk archiving can also be carried out on the Sun StorageTek 5800 system. The Sun StorageTek 5800 is an online storage appliance with an integrated hardware and software architecture in which the disk-based storage nodes are arranged in a symmetric cluster. The media abbreviation for Sun StorageTek 5800 disk archives in the `vsns` directives is `cb`.
For more information about specifying archive sets, see the `archiver.cmd`(4) man page or see the Archive Set Directives (archiver.cmd) page.

4. Save and close the `archiver.cmd` file.
5. Create a file named `diskvols.conf`.
6. Specify the directories to which the archive copies will be written.
The following example shows a `diskvols.conf` file that archives files from three archive sets. The disk volumes named `disk01` and `disk02` reside in a file system on the server system named `otherserver`. Disk volume `disk03` resides on the same host as the files to be archived.

```
# This is file sourceserver:/etc/opt/SUNWsamfs/diskvols.conf
# on the client
#
# VSN_name [host_name:] path
#
disk01 otherserver:/sam/archset1
disk02 otherserver:/sam/archset2
disk03 /sam/archset3
```

7. Save and close the `diskvols.conf` file.
8. Create directories in the file system to which the archive copies will be written. For example:

```
# mkdir sam
# cd sam
# mkdir archset1
# mkdir archset2
```

9. Verify the syntax in the `archiver.cmd` file:

```
# archiver -lv
```

10. If any errors are found, correct them before proceeding.

How to Configure Disk Archiving on the Host to Which the Archive Copies Will Be Written

Perform this procedure only if you are writing your archive copies to a host system that is different from the host system upon which the source files reside. At least one QFS or SAM-QFS file system must be created on this host. If you create source files and write archive copies to the same host system, you do not need to perform this procedure.

> Note -
> You can use the SAM-QFS Manager interface to enable disk archiving by specifying an archive policy that archives to disk VSNs. This action updates both the `archiver.cmd` file and the `diskvols.conf` file.

In this situation, you are creating a client/server environment:

- The client is the host that contains the source files.
- The server is the host to which the archive copies are written.

1. Become superuser on the server.

2. Create or open the file /etc/opt/SUNWsamfs/archiver.cmd.
3. Edit the archiver.cmd file to add disk archive set directives as in the following example:

```
#
vsns
archset1.1 dk disk01
archset2.1 dk disk02
archset3.1 dk disk03
endvsns
```

   Disk archiving can also be carried out on the Sun StorageTek 5800 system. The Sun StorageTek 5800 is an online storage appliance with an integrated hardware and software architecture in which the disk-based storage nodes are arranged in a symmetric cluster. The media abbreviation for Sun StorageTek 5800 disk archives in the vsns directives is cb.
   For more information about specifying archive sets, see the archiver.cmd(4) man page or see the Archive Set Directives (archiver.cmd) page.

4. Save and close the file.
5. Change to the file system to which you want to write the archive copies. For example:

```
# cd /ufs1
```

6. Create directories in the file system. For example:

```
# mkdir sam
# cd sam
# mkdir archset1
# mkdir archset2
```

7. Create the file /etc/opt/SUNWsamfs/diskvols.conf.
8. Specfiy the clients and endclients directives and the name of the client. The name of the client in the following example is sourceserver.

```
# This is
# file destination_server:/etc/opt/SUNWsamfs/diskvols.conf
# on the server
#
clients
sourceserver
endclients
```

9. Save and close the diskvols.conf file.

How to Enable Disk Archiving

You can enable disk archiving at any time. The procedure assumes you are adding disk archiving to an existing archiving configuration.

1. Make certain that the host to which you want to write your disk archive copies has at least one QFS or SAM-QFS file system installed on it.
2. Become superuser on the host system that contains the files to be archived.
3. Follow the How to Enable Disk Archiving on the Host That Contains the Files to Be Archived procedure or How to Configure Disk Archiving on the Host to Which the Archive Copies Will Be Written procedure.
4. On the host that contains the files to be archived, use the following command to propagate the configuration file changes and restart the system.

```
# samd config
```

5. If you are archiving to disk on a different host, follow these steps:
   a. Become superuser on the host system to which the archive copies are written.
   b. Use the samd config command to propagate the configuration file changes and restart the destination system.

6. If you are archiving to a Sun StorageTek 5800 system, you must upgrade the Sun StorageTek 5800 metadata schema configuration. Follow the procedures documented in the Sun StorageTek 5800 System Administration Guide and use the XML overlay in the following example to define the metadata that is used by SAM-QFS.

Example - Metadata Schema for SAM-QFS on an STK 5800

```
<<?xml version="1.0" encoding="UTF-8"?>

<metadataConfig>
<schema>
<namespace name="com">
<namespace name="sun">
<namespace name="samfs">
<field name="archiveId" type="string" indexable="true"/>
<field name="fileName" type="string" indexable="true"/>
<field name="modTime" type="time" indexable="true"/>
</namespace>
</namespace>
</namespace>
</schema>

<fsViews>
</fsViews>

</metadataConfig>
```

Disk Archiving Examples

The following examples show disk archiving configurations.

Example 1

In this example, VSNs identified as `disk01`, `disk02`, and `disk04` are written to `pluto`, the host system where the original source files reside. VSN `disk03` is written to a VSN on server system `mars`.

The following example shows the `diskvols.conf` file that resides on client system `pluto`.

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on pluto
# VSN Name [Host Name:]Path
#
disk01 /sam_arch1
disk02 /sam_arch2/proj_1
disk03 mars:/sam_arch3/proj_3
disk04 /sam_arch4/proj_4
```

The following example shows the `diskvols.conf` file on server system `mars`.

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on mars
#
clients
pluto
endclients
```

The following example shows a fragment of the `archiver.cmd` file on `pluto`.

```
vsns
arset1.2 dk disk01
arset2.2 dk disk02 disk04
arset3.2 dk disk03
endvsns
```

Example 2

In this example, file /sam1/testdir0/filea is in the archive set for arset0.1, and the archiver copies the content of this file to the destination path /sam_arch1.
The following example shows the diskvols.conf file.

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf
#
# VSN Name [Host Name:]Path
#
disk01 /sam_arch1
disk02 /sam_arch12/proj_1
```

The following example shows the archiver.cmd file lines that pertain to disk archiving:

```
.
vsns
arset0.1 dk disk01
endvsns .
```

The following example shows output from the sls(1) command for file filea, which was archived to disk. Note the following for copy 1:

- dk is the media type for disk archive media
- disk01 is the VSN
- f192 is the path to the disk archive tar(1) file

```
# sls -D /sam1/testdir0/filea
/sam1/testdir0/filea:
mode: -rw-r----- links: 1 owner: root group: other
length: 797904 admin id: 0 inode: 3134.49
archdone;
copy 1: ---- Dec 16 14:03 c0.1354 dk disk01 f192
access: Dec 19 10:29 modification: Dec 16 13:56
changed: Dec 16 13:56 attributes: Dec 19 10:29
creation: Dec 16 13:56 residence: Dec 19 10:32
```

Example 3

In this example, file /sam2/my_proj/fileb is on client host snickers in archive set arset0.1, and the archiver copies the content of this file to the destination path /sam_arch1 on server host mars.

The following example shows the diskvols.conf file on snickers.

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on snickers
#
# VSN Name [Host Name:]Path
#
disk01 mars:/sam_arch1
```

The following example shows the diskvols.conf file on mars.

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on mars
#
clients
snickers
endclients
```

The following example shows the directives in the archiver.cmd file that relate to this example.

```
.
vsns
arset0.1 dk disk01
endvsns .
```

Archive Directives (**archiver.cmd**)

This section provides details about the archive directives.

Global Archiving Directives

Global directives control the overall archiver operation and enable you to optimize operations for your site. You can add global directives directly to the `archiver.cmd` file, or you can specify them using the SAM-QFS Manager software. For more information on using SAM-QFS Manager to set global directives, see the SAM-QFS Manager online help.

Specify the global directives before you specify any file system directives (`fs=` directives). The archiver issues a message if it detects a global directive located after an `fs=` directive.

You can identify a global directive in the `archiver.cmd` file by either the equal sign (=) in the second field or the absence of additional fields. The following global directives are supported:

- `archivemeta` Directive
- `archmax` Directive
- `bufsize` Directive
- `drives` Directive
- `examine` Directive
- `interval` Directive
- `logfile` Directive
- `notify` Directive
- `ovflmin` Directive
- `scanlist_squash` Directive
- `setarchdone` Directive
- `wait` Directive

archivemeta Directive: Controlling Whether Metadata Is Archived

The `archivemeta` directive controls whether file system metadata is archived. If files are often moved around and there are frequent changes to the directory structures in a file system, archive the file system metadata. In contrast, if the directory structures are very stable, you can disable metadata archiving and reduce the actions performed by removable media drives. By default, metadata is not archived.

This directive has the following format:

```
archivemeta=<state>
```

For state, specify either on or `off`. The default is `off`.

The archiving process for metadata depends on whether you are using a Version 1 or a Version 2 superblock, as follows:

- For Version 1 file systems, the archiver archives directories, removable media files, segment index inodes, and symbolic links as metadata.
- For Version 2 file systems, the archiver archives directories and segment index inodes as metadata. Removable media files and symbolic links are stored in inodes rather than in data blocks. They are not archived. Symbolic links are archived as data.

archmax Directive: Controlling the Size of Archive Files

The `archmax` directive specifies the maximum size of an archive file. User files are combined to form the archive file. After the target-size value is met, no more user files are added to the archive file. Large user files are written in a single archive file.

To change the defaults, use the following directive:

```
archmax=<media> <target-size>
```

Argument	Meaning
media	The media type. For the list of valid media types, see the `mcf`(4) man page.
target-size	The maximum size of the archive file. This value is media-dependent. By default, archive files written to optical disks are no larger than 5 megabytes. The default maximum archive file size for tapes is 512 megabytes.

Setting large or small sizes for archive files has advantages and disadvantages. For example, if you are archiving to tape and `archmax` is set to a large size, the tape drive stops and starts less often. However, when writing large archive files, a premature end-of-tape causes a large amount of tape to be wasted. As a best practice, do not set the `archmax` directive to be more than 5 percent of the media capacity.

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

> Note -
> The `archmax` directive is not a valid directive for archive sets that are archived to the Sun StorageTek 5800 media type.

`bufsize` Directive: Setting the Archiver Buffer Size

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

```
bufsize=<media> <buffer-size> [lock]
```

Argument	Meaning
media	The media type. For the list of valid media types, see the `mcf`(4) man page.
buffer-size	A number from 2 through 1024. The default is 4. This value is multiplied by the dev *`blksize`* value for the media type, and the resulting buffer size is used. The _dev _`blksize` value is specified in the `defaults.conf` file. For more information about this file, see the `defaults.conf`(4) man page.
`lock`	Indicates whether the archiver can use locked buffers when making archive copies. If `lock` is specified, the archiver sets file locks on the archive buffer in memory for the duration of the `sam-arcopy`(1M) operation. This action avoids the overhead associated with locking and unlocking the buffer for each I/O request and results in a reduction in system CPU time. The `lock` argument must be specified only on large systems with large amounts of memory. Insufficient memory can cause an out-of-memory condition. The `lock` argument is effective only if direct I/O is enabled for the file being archived. By default, `lock` is not specified and the file system sets the locks on all direct I/O buffers, including those for archiving. For more information about enabling direct I/O, see the `setfa`(1) man page, the `sam_setfa`(3) library routine man page, or the `-O forcedirectio` option on the `mount_samfs`(1M) man page.

You can specify a buffer size and a lock for each archive set basis by using the archive set copy parameters, `-bufsize` and `-lock`. For more information, see Archive Set Copy Parameters.

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

By default, the archiver uses all of the drives in an automated library for archiving. To limit the number of drives used, use the `drives` directive. This directive has the following format:

```
drives=<auto-lib> <count>
```

Argument	Meaning

auto-lib	The family set name of the automated library as defined in the `mcf` file.
count	The number of drives to be used for archiving activities.

Also see the archive set copy parameters, `-drivemax`, `-drivemin`, and `-drives` described in Specifying the Number of Drives for an Archive Request: `-drivemax`, `-drivemin`, and `-drives`.

`examine` Directive: Controlling Archive Scans

New files and files that have changed are candidates for archiving. The archiver finds such files through one of the following methods:

- Continuous archiving, in which the archiver works with the file system to detect file changes immediately
- Scan archiving, in which the archiver scans the file system periodically for files that need to have changed
  This directive has the following format:

```
examine=<method>
```

method Value	Meaning
`noscan`	Specifies continuous archiving. After the initial scan, directories are scanned only when the content changes and archiving is required. Directory and inode information is not scanned. This archiving method provides better performance than scan archiving, particularly for file systems with more than 1,000,000 files. Default.
`scan`	Specifies scan archiving. The initial file system scan is a directory scan. Subsequent scans are inode scans.
`scandirs`	Specifies scan archiving on directories only. If the archiver finds a directory with the `no_archive` attribute set, the directory is not scanned. If you have files that do not change, place them in this type of directory to reduce the amount of time spent on archiving scans.
`scaninodes`	Specifies scan archiving on inodes only.

`interval` Directive: Specifying an Archive Interval

The archiver runs periodically to examine the status of all mounted archived-enabled file systems. The timing is controlled by the archive interval, which is the time between scan operations on each file system. To change the time, use the `interval` directive.

The `interval` directive initiates full scans only when continuous archiving is not set and no `startage`, `startsize`, or `startcount` parameters have been specified. If continuous archiving is set (`examine=noscan`), the `interval` directive acts as the default `startage` value. This directive has the following format:

```
interval=<time>
```

For time, specify the amount of time you want between scan operations on a file system. By default, time is interpreted in seconds and has a value of 600, which is 10 minutes. You can specify a different unit of time, such as minutes or hours.

If the archiver receives the `samu`(1M) utility's `:arrun` command, it begins scanning all file systems immediately. If the `examine=scan` directive is also specified in the `archiver.cmd` file, a scan is performed after `:arrun` or `:arscan` is issued.

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

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

`logfile` Directive: Specifying An Archiver Log File

The archiver can produce a log file that contains information about each file that is archived, re-archived, or automatically unarchived. The log file is a continuous record of archival action. By default, this file is not produced. To specify a log file, use the `logfile` directive. This directive has the following format:

```
logfile=<pathname>
```

For pathname, specify the absolute path and name of the log file. The `logfile` directive can also be set for an individual file system.

Example - Backing Up an Archiver Log File

Assume that you want to back up the archiver log file every day by copying the previous day's log file to an alternate location. Be sure to perform the copy operation when the archiver log file is closed, not while it is open for a write operation.

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

`notify` Directive: Renaming the Event Notification Script

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

```
notify=<filename>
```

For filename, specify the name of the file containing the archiver event notification script or the full path to this file. The default file name is `/etc/opt/SUNWsamfs/scripts/archiver.sh`

The archiver executes this script to process various events in a site-specific manner. The script is called with one of the following keywords for the first argument: `emerg`, `alert`, `crit`, `err`, `warning`, `notice`, `info`, and `debug`.

Additional arguments are described in the default script. For more information, see the `archiver.sh`(1M) man page.

`ovflmin` Directive: Controlling Volume Overflow

When volume overflow is enabled, the archiver can create archived files that span multiple volumes. When a file size exceeds the specified minimum size, the archiver writes the remaining portion of this file to another volume of the same type. The portion of the file written to each volume is called a section.
The `sls`(1) command lists the archive copy, showing each section of the file on each volume.

> Note -
> Use volume overflow with caution after assessing its effect on your site. Disaster recovery and recycling are much more difficult with files that span volumes. For more information, see SAM-QFS Troubleshooting and the `request`(1) man page.

The archiver controls volume overflow through the `ovflmin` directive. By default, volume overflow is disabled. To enable volume overflow, use the `ovflmin` directive in the `archiver.cmd` file. This directive has the following format:

```
ovflmin = <media> <minimum-file-size>
```

Argument	Meaning
media	The media type. For a list of valid media types, see the `mcf`(4) man page.
minimum-file-size	The minimum file size that you want to trigger the volume overflow.

The `ovflmin` directive can also be set for an individual archive set.
Volume overflow files do not generate checksums. For more information on using checksums, see the `ssum`(1) man page.

Examples of Volume Overflow

A site has many files with a significant `mo` media cartridge fraction length (such as 25 percent). These files leave unused space on each volume. To use volume space efficiently, set `ovflmin` for `mo` media to a size slightly smaller than the size of the smallest file. The following directive sets it to 150 megabytes:

```
ovflmin=mo 150m
```

In this example, two volumes are loaded for archiving and staging the files because each file overflows onto another volume.

The following example shows the archiver log file when volume overflow is enabled. The file `file50` spans three volumes with VSNs of `DLT000`, `DLT001`, and `DLT005`. The position on the volume and the size of each section is indicated in the seventh and tenth fields respectively (7eed4.1 and 477609472 for the first volume).
For a complete description of the archiver log entry, see the `archiver`(1M) man page.

```
A 97/01/13 16:03:29 lt DLT000 big.1 7eed4.1 samfs1 13.7 477609472 00 big/file50 0 0
A 97/01/13 16:03:29 lt DLT001 big.1 7fb80.0 samfs1 13.7 516407296 01 big/file50 0 1
A 97/01/13 16:03:29 lt DLT005 big.1 7eb05.0 samfs1 13.7 505983404 02 big/file50 0 2
```

This portion of the archiver log file matches the `sls -D` output for file `file50`, as shown in the following example.

```
# sls -D file50
file50:
mode: -rw-rw---- links: 1 owner: gmm group: sam
length: 1500000172 admin id: 7 inode: 1407.5
offline; archdone; stage -n
copy1: ---- Jan 13 15:55 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
```

`scanlist_squash` Directive: Controlling Scanlist Consolidation

The `scanlist_squash` parameter controls scanlist consolidation. The default setting is `off`. This parameter can be either global or file-system-specific.
When this option is enabled, the scan list entries for files in two or more subdirectories with the same parent directory that need to be scanned by `sam-arfind` at a much later time are consolidated. These directories are combined upwards to the common parent, which results in a deep recursive scan of many subdirectories. This consolidation can cause a severe performance penalty if archiving on a file system that has a large number of changes to many subdirectories.

`setarchdone` Directive: Controlling the Setting of the `archdone` Flag

The `setarchdone` parameter is a global directive that controls the setting of the `archdone` flag when the file is examined by `sam-arfind`. This directive has the following format:

```
setarchdone=on|off
```

When all archive copies for a file have been made, the `archdone` flag is set for that file to indicate that no further archive action is required. During directory scans, the `archdone` flag is also set for files that will never be archived. Because evaluating whether a file will ever be archived can affect performance, the `setarchdone` directive gives you control over this activity. This directive controls the setting of the `archdone` flag only on files that will never be archived. It does not affect the setting of the `archdone` flag after archive copies are made.
The default setting for the directive is `off` if the `examine` directive is set to `scandirs` or `noscan`.

`wait` Directive: Delaying Archiver Startup

The `wait` directive causes the archiver to wait for a start signal from `samu`(1M) or SAM-QFS Manager. By default, the archiver begins archiving when started by `sam-fsd`(1M). This directive has the following format:

```
wait
```

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

File System Directives

Archiving controls apply to all file systems. However, you can confine some controls to an individual file system. When the archiver encounters an `fs=` directive in the `archiver.cmd` file, all subsequent directives are applied to that specific file system. Therefore, in the `archiver.cmd` file, place any `fs=` directives after the general directives.

You can specify `fs=` directives either by editing the `archiver.cmd` file as described in the following sections, or by using the SAM-QFS Manager software. See the SAM-QFS Manager online help for more information.

`fs` Directive: Specifying the File System

Use the `fs=` directive to specify actions for a particular file system.
For instance, you can use this directive to create a different log file for each file system. This directive has the following format:

```
fs=<fsname>
```

For fsname, specify the file system name as defined in the `mcf` file.

The general directives and archive set association directives that occur after a `fs=` directive apply only to the specified file system until another `fs=` directive is encountered.

Global Directives As File System Directives

Several directives can be specified both as global directives for all file systems and as directives specific to only one file system:

- `interval` directive
- `logfile` directive
- `scanlist_squash` directive
- `wait` directive

Archive Copy Directives

By default, the archiver writes a single archive copy for files in the archive set when the archive age of the file is four minutes. To change the default behavior, use archive copy directives. Archive copy directives must appear immediately after the archive set assignment directive to which they pertain.

The archive copy directives begin with a copy-number value of 1, 2, 3, or 4. The digit is followed by one or more arguments that specify archive characteristics for that copy. Each archive copy directive has the following format:

```
<copy-number> [ -release | -norelease ] [<archive-age>] [<unarchive-age>]
```

You can specify archive copy directives either by editing the `archiver.cmd` file as described in the following sections, or by using the SAM-QFS Manager software. For more information, see the SAM-QFS Manager online help.

The following sections describe the archive copy directive arguments.

`-release` Directive: Releasing Disk Space After Archiving

To specify that the disk space for files is released after an archive copy is made, use the `-release` directive after the copy number. This directive has the following format:

```
-release
```

Example - ***archiver.cmd*** File Using the ***-release*** Directive

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

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

`-norelease` Directive: Delaying Disk Space Release

The `-norelease` option prevents the automatic release of disk cache until all copies marked with `-norelease` are made. The `-norelease` directive makes the archive set eligible to be released after all copies have been archived, but the files are not released until the releaser is invoked and selects them as release candidates. This directive has the following format:

```
-norelease
```

Using the `-norelease` directive on a single copy has no effect on automatic releasing.

Example - ***archiver.cmd*** File Using the ***-norelease*** Directive

The following example specifies an archive set named `vault_tapes`. Two copies are created and then the disk cache associated with this archive set is released.

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

Using **`-release`** and **`-norelease`** Together

To make sure that disk space is released immediately after all copies of an archive set have been archived, use the `-release` and `-norelease` directives together. The combination of `-release` and `-norelease` causes the archiver to release the disk space immediately after all copies having this combination are made, rather than waiting for the releaser to be invoked.

Setting the Archive Age

Change the timing for archiving files by specifying the archive age. Specify the time with a suffix character such as `h` for hours or `m` for minutes as shown in Table - File Age Suffixes.

Example - ***archiver.cmd*** File That Specifies the Archive Age

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

```
ex_set data
1 1h
```

Unarchiving Automatically

If you specify more than one archive copy of a file, you can unarchive all but one of the copies automatically. You can do this when the files are archived to various media using various archive ages.

Example - ***archiver.cmd*** File that Specifies the Unarchive Age

The following example shows the directive that specifies the unarchive age. The first copy of the files in the path `home/users` is archived six minutes after modification. When the files are 10 weeks old, the archiver creates the second and third archive copies and unarchives the first copy.

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

For more ways to control unarchiving, see Controlling Unarchiving.

Specifying More Than One Copy for Metadata

If more than one copy of metadata is required, place copy definitions in the `archiver.cmd` file immediately after the `fs=` directive.

Example - ***archiver.cmd*** File that Specifies Multiple Metadata Copies

In this example, one copy of the metadata for the `samfs7` file system is made after 4 hours and a second copy is made after 12 hours.

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

File system metadata includes path names in the file system. If you have frequent changes to directories, the new path names cause the creation of new archive copies and results in frequent loading of the volumes specified for metadata.

Archive Set Directives (archiver.cmd)

This section provides information about the archive set directives:

- About Archive Set Directives
- Archive Set Copy Parameters
- VSN Association Directives
- VSN Pools Directives

About Archive Set Directives

The following archive sets are available by default:

- Reserved archive set: `no_archive` and `allsets`.
  - The `no_archive` archive set is defined by default. Files assigned to this archive set are never archived. For example, files in a temporary directory can be assigned to the `no_archive` archive set.
  - The `allsets` archive set defines parameters that apply to all archive sets.
- Each file system has a default archive set with the same name that cannot be changed. These archive sets are reserved for control structure information. For each file system, both the metadata and data files are archived. The file system archive set includes the directory and link information and any files that are not included in another archive set.

By default, files are archived as part of the archive set named for the file system. However, you can create archive sets for files that share common criteria such as size, ownership, group, or directory location. If a file does not match one of the specified archive sets, it is archived as part of the default archive set. A file in the file system can be a member of only one archive set. All files in an archive set are copied to the volumes associated with that archive set.

Archive files are compatible with the standard UNIX `tar`(1) format for data compatibility with the Solaris OS and other UNIX systems. If a complete loss of your SAM-QFS environment occurs, the `tar`(1) format allows file recovery using standard UNIX tools and commands.

The characteristics of archive set are controlled by the archive set directives. These directives are arranged in the following categories:

- Assignment directive define archive sets: the destination of the archive copy, how long the copy is kept archived, and how long the software waits before archiving the data
- Copy parameters define how each archive set is archived: The archiving process copies the data necessary for file system operations, including directories, symbolic links, the index of segmented files, and archive media information.
- VSN association directives assign volumes to archive sets.
- VSN pools directives define a collection of volumes.

You can create archive sets either by editing the `archiver.cmd` file as described in the following sections, or by using the SAM-QFS Manager software. In the SAM-QFS Manager, an archive policy defines an archive set. For more information, see the SAM-QFS Manager online help.

Archive Set Assignment Directive

The archive set assignment directive select files according to shared characteristics to include in archive sets. Each archive set assignment directive has the following format:

```
<archive-set-name> <path> [<search-criterion1> <search-criterion2> ... ] [<file-attribute1>
<file-attribute2> ...]
```

Argument	Meaning
archive-set-name	A site-defined name for the archive set. A best practice is to define a name that identifies the common characteristics of the files belonging to the archive set. The name has the following requirements: • Maximum of 29 characters • Uppercase and lowercase alphabetic characters, numbers 0-9, and the underscore character (_). • No other special characters or spaces are allowed. • The first character must be a letter. • You cannot create an archive set with the name of one of the reserved archive sets, `no_archive` or `allsets`. To prevent archiving of the files in an archive set, specify_ the name as *`no_archive`*
path	Specify the path relative to the mount point of the file system so that files in the directory specified by path and its subdirectories are included in this archive set. To include all of the files in a file system, use a period (.). A leading slash (/) is not allowed in the path.
search-criterion1 search-criterion2	Zero, one, or more search criteria can be specified to restrict the archive set to files that meet the criteria. Search criteria include file age, file size, file ownership, and file name.
file-attribute1 file-attribute2	Zero, one, or more file attributes can be specified. These file attributes are set for files as the `sam-arfind` process scans a file system during archiving.

Examples - Archive set assignment directives.

In this example, all files in the user account `hmk` are archived in a separate archive set. All files over 1 MB in size in the directories in the `xray` group are archived in the `datafiles` archive set. All other files are archived in the `system` archive set.

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

The following example shows directives that prevent archiving of files in a `tmp` directory at any level and regardless of the directory in which the `tmp` directory resides within the file system.

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

File Age search-criterion: `-access` and `-nftv`

To use the last time a file was opened to define assignment to an archive set, use the `-access` age characteristic as one of the search-criterion arguments.

This characteristic causes files that have not been accessed within the value of age to be re-archived to different, less-expensive media. For age, specify an integer followed by one of the suffixes shown in the following table.

Table - File Age Suffixes

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

When determining age, the software validates the access and modification times for files to ensure that these times are greater than or equal to the file creation time, and less than or equal to the time at which the file is examined. For files that have been migrated into a directory, this validation might not result in the desired behavior. Use the `-nftv` (no file time validation) parameter in these situations to prevent the validation of file access and modification times.

File Age search-criterion: `-after`

Use the `-after` date-time characteristic to include files that have been modified or created recently into the same archive set. Only files created or modified after the date indicated are included in the archive set. Specify the date and time in the following format:

```
<YYYY>-<MM>-<DD>[T<hh>:<mm>:<ss>][Z]
```

If the time is not specified, the default time is 00:00:00. If the `Z` is included, the time is Coordinated Universal Time (UTC). If the `Z` is not included, the time is local time.

File Size search-criterion: `-minsize` and `-maxsize`

Use the `-minsize` size and `-maxsize` size characteristics to restrict membership in an archive set to those over or under a specified size. For size, specify an integer followed by one of the letters shown in the following table.

Table - ***`-minsize`*** and ***`-maxsize`*** size Suffixes

Letter	Meaning
`b`	Bytes
`k`	Kilobytes
`M`	Megabytes
`G`	Gigabytes
`T`	Terabytes
`P`	Petabytes
`E`	Exabytes

Example - Using the ***`-minsize`*** and ***`-maxsize`*** Characteristics

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

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

Owner and Group search-criterion: **-user** and **-group**

To restrict membership in an archive group to ownership and group affiliation, use the `-user` name and `-group` name characteristics.

Example - Using the ***-user*** and ***-group*** Directive

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

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

File Name search-criterion Using Pattern Matching: **-name** regex

To specify that file names are used for assignment to an archive set, use `-name` regex characteristic, which specifies that any complete path matching the regular expression regex is to be a member of the archive set.

All files beneath the selected directory (with their specified paths relative to the mount point of the file system) go through pattern matching. Therefore, you can specify patterns in the `-name` regex field to match both file names and path names.

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

Examples - ***-name***

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

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

The following directive selects files that start with the characters GEO for the `satellite` archive set:

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

The following directive prevents any file ending with `.o` from being archived:

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

Example 1 - Pattern Matching with Regular Expression

The archive set assignment directive in the following example does not archive `fred.*` in the user directories or subdirectories.

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

As a result, the following files are not archived:

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

The following files are archived:

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

Example 2 - Pattern Matching with Regular Expression

The archive set assignment directive in the following example does not archive `fred.*` in the user home directories but it does archive `fred.*` in the user subdirectories and in the directory `share/marketing`. In this case, a user home directory is anything from `share/marketing/` until the next slash character "/".

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

The following files are not archived:

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

The following files are archived:

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

Release and Stage file-attributes: -release and -stage

You can set the release and stage attributes associated with files within an archive set by using the `-release` and `-stage` options, respectively. Both of these settings override any existing stage or release attributes.

See How to Specify Release Attributes for All Files in an Archive Set for information about the `-release` attribute.

The `-stage` option has the following format:

```
-stage <attribute>
```

Attribute	Meaning
a	Stage the files in this archive set associatively.

d	Reset to default.
n	Never stage the files in this archive set.

Example - Archive Sets and File Attributes

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

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

Membership Conflicts in Archive Sets

When the selection of a file for inclusion in an archive set is ambiguous, the archiver uses the following rules:

- The membership definition occurring first in the archive set is chosen. Place the most restrictive assignment directives early in the `archiver.cmd` file.
- Membership definitions local to a file system are chosen before any global definitions.
- A membership definition that exactly duplicates a previous definition is noted as an error.

The archiver evaluates the file-system-specific directives before evaluating the global directives. Therefore, files can be assigned to a local archive set (including the `no_archive` archive set) instead of being assigned to a global archive. This result has implications for global archive set assignments such as `no_archive`.

Example - ***`archiver.cmd`*** File With Membership Conflicts

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

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

Example - ***`archiver.cmd`*** File Without Membership Conflicts

The following example shows the directives to use to ensure that no `.o` files are archived in the two file systems.

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

Archive Set Copy Parameters

The archive set copy parameters define how each archive set is archived: the data files, directories, symbolic links, the index of segmented files, and archive media information.
The archive set copy parameters section of the `archiver.cmd` file begins with the `params` directive and ends with the `endparams` directive.
The following example shows the format for copy parameters for an archive set.

```
params
<archive-set-name>.<copy-number>[R] [<-param1> <-param2> ...]
.
.
.
endparams
```

Table - Arguments for the Archive Set Copy Parameters

Argument	Meaning
archive-set-name	A site-defined name for the archive set. A best practice is to define a name that identifies the common characteristics of the files belonging to the archive set. The name has the following requirements: • Maximum of 29 characters • Uppercase and lowercase alphabetic characters, numbers 0-9, and the underscore character (_). • No other special characters or spaces are allowed. • The first character must be a letter.
.	A period (`.`) character. Used to separate archive-set-name from copy-number.
copy-number	An integer that defines the archive copy number: 1, 2, 3, or 4.
R	Specifies that the parameters being defined are for re-archived copies of this archive set. For example, you can use the R and specify VSNs in the -param1 argument to direct re-archived copies to specific volumes.
-param1 -param2	One or more parameters such as maximum size, buffer size, number of drives, and son on. The following subsections describe the parameters than can be specified between the `params` and `endparams` directives.

To set default directives for all archive sets, specify directives for the archive set `allsets` archive set. The `allsets` directives must precede the directives for archive set copies because parameters set for individual archive set copies override parameters set for the `allsets` directive. For more information about the `allsets` archive set, see the `archiver.cmd`(4) man page.

You can specify archive set copy parameters by editing the `archiver.cmd` file as described in the following sections or by using the SAM-QFS Manager software. For more information, see the SAM-QFS Manager online help.

The following subsections describe all archive set processing parameters, with the exception of disk archiving parameters. For information about disk archiving parameters, see About Disk Archiving.

Controlling the Size of Archive Files: -archmax

The `-archmax` parameter sets the maximum file size for an archive set. This parameter has the following format:

```
-archmax <target-size>
```

This parameter is very similar to the `archmax` global directive. For information about that directive and the values to enter for target-size, see archmax Directive: Controlling the Size of Archive Files.

Setting the Archiver Buffer Size: -bufsize

By default, a file being archived is stored in memory in a buffer of a default size for the media type before being written to archive media. Use the `-bufsize` directive to specify a buffer size. A custom size can improve performance. This parameter has the following format:

```
-bufsize=<buffer-size>
```

The default buffer size is 4, indicating that the actual buffer size is 4 multiplied by the `devblksize` value for the media type. Specify a number from 2 to 32. The `devblksize` value is specified in the `defaults.conf` file.
For more information about this file, see the `defaults.conf`(4) man page.

Example - Buffer Size: -bufsize

```
myset.1 -bufsize=6
```

This parameter is similar to the `bufsize`=media buffer-size global directive. For more information about that directive, see bufsize Directive: Setting the Archiver Buffer Size.

Specifying the Number of Drives for an Archive Request: -drivemax, -drivemin, and -drives

By default, the archiver uses one media drive to archive the files of one archive set. When an archive set has many files or large files, using more than one drive is advantageous. In addition, if the drives in your automated library operate at different speeds, use of multiple drives can balance these variations and increase archiving efficiency. The drive directives have the following formats:

```
-drivemax <max-size>
-drivemin <min-size>
-drives <number>
```

Argument	Meaning
max-size	The maximum amount of data to be archived using one drive.
min-size	The minimum amount of data to be archived using one drive. The default is the `-archmax` target-size value (if specified) or the default value for the media type. If you specify the `-drivemin` min-size directive, the SAM-QFS software uses multiple drives only when enough activity occurs to warrant it. As a guideline, set min-size to be large enough to cause the transfer time to be significantly longer than the cartridge change time (load, position, unload).
number	The number of drives to be used for archiving this archive set. The default is 1.

An archive request is evaluated against the parameters that are specified, as follows:

- If an archive request is less than the value of min-size, only one drive is used to write an archive request.
- If an archive request is larger than the value of min-size, the archive request is evaluated against min-size and the appropriate number of drives is scheduled up to the full number of drives specified.
- If the value of min-size is 0, an attempt is made to split the archive request among the full number of drives specified.

When you use the `-drives` parameter, multiple drives are used only if data that is more than the value of min-size is to be archived. The number of drives to be used in parallel is the lesser of the following two values:

- The size of the archive request divided by the value of min-size
- The number of drives specified by the `-drives` parameter

Use the `-drivemin` and `-drives` parameters when you want to divide an archive request among drives but do not want to have all the drives busy with small archive requests. This situation can occur with very large files.

To set these parameters, consider file creation rates, the number of drives, the time it takes to load and unload drives, and drive transfer rates. For example, a site splits an archive set named `bigfiles` across five drives. This archive set could be split as shown in the following table.

Archive Set Size	Number of Drives
< 20 gigabytes	1
> 20 gigabytes to < 30 gigabytes	2
> 30 gigabytes to < 40 gigabytes	3
> 40 gigabytes to < 50 gigabytes	4
> 50 gigabytes	5

Example - Directives Used to Split an Archive Request Over Multiple Drives

The following example how to split the archive requests of 10 GB or more over five drives.

```
params
bigfiles.1 -drives 5 -drivemin 10G
endparams
```

In addition, the following line ensures that two drives are used to archive the files when the total size of the files in archive set `huge_files.2` is equal to or greater than two times `drivemin` for the media.

```
huge_files.2 -drives 2
```

Maximizing Space on a Volume: -fillvsns

By default, the archiver selects a volume with enough space for all files when it writes an archive copy. This action results in volumes not being filled to capacity. When `-fillvsns` is specified, the archiver separates the archive request into smaller groups and can use different volumes.

Specifying Archive Buffer Locks: -lock

By default, a file is stored in a buffer before being written to archive media. If direct I/O is enabled, you can use the `-lock` parameter to lock this buffer. The `-lock` parameter indicates that the archiver must use locked buffers when making archive copies. If `-lock` is specified, the archiver sets file locks on the archive buffer in memory for the duration of the `sam-arcopy`(1M) operation. This action avoids paging of the buffer, and can improve performance.

This parameter has the following format:

```
-lock
```

Use the `-lock` parameter only on large systems with large amounts of memory. Insufficient memory can cause an out-of-memory condition.

The `-lock` parameter is effective only if direct I/O is enabled for the file. By default, `-lock` is not specified, and the file system sets locks on all direct I/O buffers, including those for archiving. For more information about enabling direct I/O, see the `setfa`(1) man page, the `sam_setfa`(3) library routine man page, or the `-O forcedirectio` option on the `mount_samfs`(1M) man page.

This paramter is similar to the `lock` argument to the `bufsize` global directive. For more information about this topic, see bufsize Directive: Setting the Archiver Buffer Size.

Making Archive Copies of Offline Files: -offline_copy

A file is a candidate for being released after one archive copy is made. If the file is released and goes offline before any remaining archive copies are made, the archiver uses this parameter to determine the method use to make the other archive copies. When you specify the method, consider the number of drives available to the SAM-QFS system and the amount of disk cache available. This parameter has the following format:

```
-offline_copy <method>
```

method Value	Meaning
none	Stages files as needed for each file before copying to the archive volume. Default.
`direct`	Copies files directly from the offline volume to the archive volume without using the cache. This method assumes that the source volume and the destination volume are different volumes and that two drives are available. Raise the value of the `stage_n_window` mount option to a value that is greater than its default of 256 kilobytes. For more information about mount options, see the `mount_samfs`(1M) man page.
`stageahead`	Stages the next archive file while writing an archive file to its destination.
`stageall`	Stages all files to disk cache before archiving. This method uses one drive and assumes that room is available on disk cache for all files.

Specifying Recycling

Use the recycling process to reclaim space on archive volumes in use by expired archive images. By default, no recycling occurs. You must specify directives in both the `archiver.cmd` file and the `recycler.cmd` file. For more information, see the Configuring the Recycler page.

Sorting Archive Files: -sort and -rsort

By default, files in an archive set are sorted by path before they are archived. You can specify that files are sorted by `age`, `priority`, `size` or not sorted (none). Only one sort method can be used per archive set.
You can use `-rsort` to reverse the order of sorting specified by method.

Example - Sorting Files in an Archive Set

The first example line sorts the archive set copy `cardiac.2` by the age of the file, oldest to youngest. The second line forces the archive set copy `catscans` to be sorted by the size of the file, in reverse order, largest to smallest.
`size`.

```
cardiac.2 -sort age
catscans.3 -rsort size
```

Controlling Unarchiving

Unarchiving is the process by which archive entries for files or directories are deleted. A file is unarchived based on the time since it was last accessed. This distinction means data that is accessed frequently can be stored on fast media such as disk and infrequently-accessed data can be stored on tape. By default, files are never unarchived.

Example - Directives to Control Unarchiving

This following example directives specify that the {{arset1}}file remains on disk all the time, even if it is older than 60 days. The Copy 1 information is removed when the file has not been accessed for 60 days. After the Copy 1 information is removed, any access request is fulfilled by Copy 2 and is read from tape. The archiver makes a new Copy 1 on disk and the 60-day cycle starts again.

```
arset1 dir1
1 10m 60d
2 10m
3 10m
vsns
arset1.1 mo OPT00[0-9]
arset1.2 lt DLTA0[0-9]
arset1.3 lt DLTB0[0-9]
```

The example directives meet the requirements for both access and archiving in the following scenario.
A patient is in the hospital for four weeks. During this time, all the patient's files are on fast media and the data is being access frequently. This is Copy 1 (copy 1=mo). After two weeks, the patient is discharged from the hospital. The patient files are accessed less frequently and then not at all. When no data has been accessed for this patient 60 days, the Copy 1 entry in the inode is unarchived. Only Copy 2 and Copy 3 entries are available. The volume of fast media can now be recycled and used by current patients without having to increase the disk library. However, six months later, the patient returns to the hospital. The first access of the patient's file is from tape (Copy 2). To get the data on fast media, the archiver creates a new Copy 1 on disk, ready for new information.

Controlling How Archive Files Are Written: -tapenonstop

By default, the archiver writes a tape mark, an end of file (EOF) label, and two more tape marks between archive files. When the next archive file is started, the driver backs up to the position after the first tape mark, causing a loss of performance. The `-tapenonstop` parameter directs the archiver to write only the initial tape mark. In addition,the archiver enters the archive information at the end of the copy operation.

For more information about the `-tapenonstop` parameter, see the `archiver.cmd`(4) man page.

Reserving Volumes: -reserve

By default, the archiver writes archive set copies to any volume specified by a regular expression as described in the volume associations section of the archiver.cmd file. However, if you require that an archive set volume contains files from only one archive set, you can reserve a volume for this purpose.

> Note -
>
> - A site that uses reserved volumes incurs more cartridge loads and unloads.
> - A site that uses reserved volumes for file systems that have many directories of a few small files causes the archiver to write many small archive files to each reserved volume. These small archive files, each with its own tar(1) header, slow performance.

The -reserve parameter specifies a volume for use by an archive set and gives it a unique identifier that ties the archive set to the volume. The volume identifier is not assigned to any other archive set copy, even if a regular expression matches it. The format for the -reserve parameter is as follows:

```
-reserve <keyword>
```

The value of keyword depends on the form you are using. You can specify one, two, or all three forms in combination.

Form	keyword	Reserved Name Examples	Notes
Archive Set	set	users.1//Data.1//	
Owner	dir	proj.1/p105/ proj.1/p104/	The dir, user, and group keywords, which are mutually exclusive, specify the owner component in the reserved name. The dir keyword uses the directory path component immediately following the path specification of the archive set definition.
	user	users.1/user5/ users.1/user4/	
	group	data.1/engineering/	
File System	fs	proj.1/p103/samfs1 proj.1/p104/samfs1	The fs keyword specifies the file system component in the reserved name.

Example - Reserving Volumes by Archive Set

The following example specifies that the allsets archive set reserves a volume for each archive set.

```
params
allsets -reserve set
endparams
```

Example - Reserved Volume Name

The following example specifies that the arset.1 archive set reserves a volume and the volume identifier is created from an archive set, a group, and the file system.

```
params
arset.1 -reserve set -reserve group -reserve fs
endparams
```

Information about reserved volumes is stored in the library catalog. The lines in the library catalog that describe reserved volumes begin with #R characters and show the media type, the VSN, the reserve information, and the reservation date and time. The information also includes the archive set component, path name component, and file system component, separated by two slashes (//).

> **Note -**
> The slash characters do not indicate a path name. They serve to separate the components of a reserved name.

Example - Library Catalog Showing Reserved Volumes

The lines have been truncated to fit on the page.

```
6 00071 00071 lt 0xe8fe 12 9971464 1352412 0x6a000000 131072 0x
# -il-o-b----- 05/24/00 13:50:02 12/31/69 18:00:00 07/13/01 14:03:00
#R lt 00071 arset0.3// 2001/03/19 18:27:31
10 ST0001 NO_BAR_CODE lt 0x2741 9 9968052 8537448 0x68000000 1310
# -il-o------- 05/07/00 15:30:29 12/31/69 18:00:00 04/13/01 13:46:54
#R lt ST0001 hgm1.1// 2001/03/20 17:53:06
16 SLOT22 NO_BAR_CODE lt 0x76ba 6 9972252 9972252 0x68000000 1310
# -il-o------- 06/06/00 16:03:05 12/31/69 18:00:00 07/12/01 11:02:05
#R lt SLOT22 arset0.2// 2001/03/02 12:11:25
```

One or more of the reserve information fields can be empty, depending on the options defined in the `archiver.cmd` file. A reservation line is appended to the file for each volume that is reserved for an archive set during archiving.

You can also use the `reserve`(1M) and `unreserve`(1M) commands to reserve and unreserve volumes. For more information about these commands, see the `reserve`(1M) and `unreserve`(1M) man pages.
A volume is unreserved when it is relabeled because the archive data has been effectively erased.

You can display the reserve information by using the `samu`(1M) utility's v display or by using the `archiver`(1M) or `dump_cat`(1M) command in one of the formats shown in the following example.

Example - Commands to Use to Display the Reserve Information

```
archiver -lv
dump_cat -V _catalog-name_
```

Example: User and Data Files Archived to Optical Media shows a complete archive example using reserved volumes.

Setting Archive Priorities: -priority

Archive-enabled file systems provide priorities for archiving files. Each file is assigned a priority computed from properties of the file and priority multipliers that can be set for each archive set in the `archiver.cmd` file. Properties include online/offline, age, number of copies made, and size.

By default, the files in an archive request are not sorted, and all property multipliers are zero. The result is that files are archived in first-found, first-archived order. To change the order in which files are archived, set priorities and sort methods. Examples of new priorities include:

- Select the `priority` sort method to archive files within an archive request in priority order.
- Change the `archive_loaded` priority to reduce media loads.
- Change the `offline` priority to cause online files to be archived before offline files.
- Change the `copy#` priorities to make archive copies in copy order.

Table - Archive Priorities

Archive Priority	Definition
`-priority age` value	Archive age property multiplier
`-priority archive_immediate` value	Archive immediate property multiplier
`-priority archive_overflow` value	Multiple archive volumes property multiplier
`-priority archive_loaded` value	Archive volume loaded property multiplier
`-priority copies` value	Copies-made property multiplier

`-priority copy1` value	Copy 1 property multiplier
`-priority copy2` value	Copy 2 property multiplier
`-priority copy3` value	Copy 3 property multiplier
`-priority copy4` value	Copy 4 property multiplier
`-priority offline` value	File offline property multiplier
`-priority queuewait` value	Queue wait property multiplier
`-priority rearchive` value	Rearchive property multiplier
`-priority reqrelease` value	Reqrelease property multiplier
`-priority size` value	File-size property multiplier
`-priority stage_loaded` value	Stage volume loaded property multiplier
`-priority stage_overflow` value	Multiple stage volumes property multiplier

For value, specify a floating-point number in the following range:

```
-3.400000000E+38 <= _value_ <= 3.402823466E+38
```

For more information about priorities, see the `archiver`(1M) and `archiver.cmd`(4) man pages.

Scheduling Archiving: -startage, -startcount, and -startsize

As the archiver scans a file system, it identifies files to be archived. Files that are recognized as candidates for archiving are placed in a list known as an archive request. At the end of the file system scan, the system schedules the archive request for archiving. The `-startage`, `-startcount`, and `-startsize` archive set parameters control the archiving workload and ensure the timely archival of files.

Table - Formats for the ***-startage***, ***-startcount***, and ***-startsize*** Directives

Directive	Meaning
`-startage` time	The amount of time that can elapse between the first file in a scan being marked for inclusion in an archive request and the start of archiving. Specify a time in the format used in Setting the Archive Age. If this variable is not set, the `interval` directive is used.
`-startcount` count	The number of files to be included in an archive request. When the number of files in the archive request reaches the this value, archiving begins. By default, count is not set.
`-startsize` size	The minimum total size, in bytes, of all files to be archived in an archive request. Archiving work is accumulated, and archiving begins when the total size of the files reaches the this value. By default, size is not set.

The `examine`=method directive and the `interval`=time directives interact with the `-startage`, `-startcount`, and `-startsize` directives. The `-startage`, `-startcount`, and `-startsize` directives optimally balance archive timeliness and archive work done. These values override the `examine`=method specification, if any. For more information about `examine` directive, see examine Directive: Controlling Archive Scans. For more information about the `interval` directive, see interval Directive: Specifying an Archive Interval.

The `-startage`, `-startcount`, and `-startsize` directives can be specified for each archive copy. If more than one of these directives is specified, the first condition encountered starts the archive operation. If none of these directives is specified, the archive request is scheduled based on the `examine`=method directive:

- If `examine=noscan`, the default values of the directives are used: `startage` 10 minutes, `startcount` 10,000, and `startsize` 10 gigabytes. The archive request is scheduled according to the value of the `interval=` directive after the first file is entered in the archive request. This method is continuous archiving and is the default method.
- If `examine=scan|scaninodes|scandirs`, the archive request is scheduled for archiving after the file system scan.

The `archiver.cmd`(4) man page provides examples that show how to use these directives.

VSN Association Directives

The VSN associations section of the `archiver.cmd` file assigns volumes to archive sets. This section starts with a `vsns` directive and ends with an `endvsns` directive.

VSN associations can also be configured with the SAM-QFS Manager software. See the SAM-QFS Manager online help for more information.

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

```
<archive-set-name>.<copy-num> <media-type> <vsn-expr> ... [-pool <vsn-pool-name> ...]
```

An association requires at least three fields: archive-set-name and copy-num, media-type, and at least one volume. The archive-set-name and copy-num values are connected by a period (.).

Table - Arguments for the VSN Association Directive

Argument	Meaning
archive-set-name	A site-defined name for the archive set.
copy-num	A digit followed by one or more arguments that specify archive characteristics for that copy. Archive copy directives begin with a digit. This digit (1, 2, 3, or 4) is the copy number.
media-type	The media type. For a list of valid media types, see the `mcf`(4) man page.
vsn-expr	A regular expression. See the `regexp`(5) man page.
`-pool` vsn-pool-name	A named collection of VSNs.

> Note -
> If your SAM-QFS environment is configured to recycle by archive set, do not assign a VSN to more than one archive set.

Example - VSN Specifications on Multiple Lines

The following example shows two lines of VSN specifications.

```
vsns
set.1 lt VSN001 VSN002 VSN003 VSN004 VSN005
set.1 lt VSN006 VSN007 VSN008 VSN009 VSN010
endvsns
```

Example - VSN Specifications With a Continued Line

The following example shows a VSN specification that uses a backslash character "\" to continue a line onto a subsequent line.

```
vsns
set.1 lt VSN001 VSN002 VSN003 VSN004 VSN005 \
VSN006 VSN007 VSN008 VSN009 VSN010
endvsns
```

Example - VSN Specifications With Shorthand Notation

The following example specifies VSNs using a regular expression in a shorthand notation.

```
vsns
set.1 lt VSN0[1-9] VSN10
endvsns
```

When the archiver needs volumes for the archive set, it examines each volume of the selected media type in all automated libraries and manually mounted drives to determine whether the volume satisfies any VSN expression. It selects the first volume that matches an expression that contains enough space for the archive copy operation. For example:

- The following directive specifies that files belonging to archive set `ex_set` for copy 1 be copied to media type `mo` using any of the 20 volumes with the names `optic20` through `optic39`.

```
ex_set.1 mo optic[2-3][0-9]
```

- The following directive specifies that files belonging to archive set `ex_set` for copy 2 be copied to media type `lt` using any volume beginning with TAPE:

```
ex_set.2 lt ^TAPE
```

> Note -
> Make sure you assign volumes to the archive set used for the file system's metadata when setting up the `archiver.cmd` file. Each file system has an archive set with the same name as the file system. For more information about preserving metadata, see the `samfsdump`(1M) man page or see the SAM-QFS Troubleshooting page.

VSN Pools Directives

The VSN pools section of the `archiver.cmd` file starts with a `vsnpools` directive and ends either with an `endvsnpools` directive or with the end of the `archiver.cmd` file. This section names a collection of volumes.

VSN pools can also be configured with the SAM-QFS Manager software. See the SAM-QFS Manager online help for more information.

A VSN pool is a named collection of volumes. VSN pools are useful for defining volumes that can be available to an archive set. As such, VSN pools provide a useful buffer for assigning volumes and reserving volumes to archive sets. You can use VSN pools to define separate groups of volumes by departments within an organization, by users within a group, by data type, and according to other convenient groupings.

If a volume is reserved, it is no longer available to the pool in which it originated. Therefore, the number of volumes within a named pool changes as volumes are used. You can view the VSN pools by issuing the `archiver`(1M) command in the following format:

```
# archiver -lv | more
```

The syntax of a VSN pool definition is as follows:

```
<vsn-pool-name> <media-type> <vsn-expr>
```

Argument	Meaning
vsn-pool-name	The VSN pool.
media-type	The two-character media type. For a list of valid media types, see the `mcf`(4) man page.
vsn-expr	A regular expression. You can provide one or more vsn-expr arguments. See the `regcmp`(3G) man page.

The following example uses four VSN pools: `users_pool`, `data_pool`, `proj_pool`, and `scratch_pool`. A scratch pool is a set of volumes used when specific volumes in a VSN association are exhausted or when another VSN pool is exhausted. If one of the three specific pools is out of volumes, the archiver selects the scratch pool VSNs.

Example - VSN Pools

The following example shows an `archiver.cmd` file that uses four VSN pools.

```
vsnpools
users_pool mo ^MO[0-9][0-9]
data_pool mo ^DA.*
scratch_pool mo ^SC[5-9][0-9]
proj_pool mo ^PR.*
endvsnpools
vsns
users.1 mo -pool users_pool -pool scratch_pool
data.1 mo -pool data_pool -pool scratch_pool
proj.1 mo -pool proj_pool -pool scratch_pool
endvsns
```

For more information about VSN associations, VSN Association Directives

About Releasing

This section describes the releasing process and releaser operations.

Releasing is the process by which the releaser makes disk cache space available by identifying archived files and releasing their disk cache copy. This action makes room for other files to be created or staged from archive media. The releaser can release only archived files. A released file has no data on the disk cache.

The Sun Storage Archive Manager (SAM) software invokes the releaser process when a site-specified disk threshold is reached. As an alternative, you can use the `release`(1) command to release a file's disk space immediately or to set releasing parameters for a file.

You can also specify that files are released immediately after archiving, that files are never released, or that files are partially released.

Releaser Process

When file system utilization exceeds its configured high-water mark, the file system management software invokes the releaser, which does the following:

- Reads the `releaser.cmd` file and collects the directives that control the release process
- Scans the file system and collects information about each file
- Begins releasing files in priority order

A file system can contain thousands of files. Keeping track of the release priority for every file can be wasteful, because releasing only several large files might return the file system to its low-water mark. However, the releaser must examine the priority of each file or risk missing the best candidates for release. The releaser does this by identifying the first 10,000 candidates. It then discards subsequent candidates if they do not have a priority greater than the lowest-priority candidate among the first 10,000.

After the releaser has determined the priority of the first 10,000 candidates, it selects the files with the highest priority for release. After releasing each file, the releaser checks whether the file system cache utilization is below the low-water mark. If so, it stops releasing files. If not, it continues releasing the files in priority order.

If the releaser has released all 10,000 candidates and the file system is still above the low-water mark, it starts over and identifies 10,000 new candidates.

The releaser exits if it cannot find any viable candidates. This situation can occur, for example, if files do not yet have archive copies. In this case, the SAM software starts the releaser again after one minute has elapsed.

The high and low-water marks are set with the `high`=percent and `low`=percent file system mount options. For more information about these mount options, see the `mount_samfs`(1M) man page.

Releaser Concepts

This section describes concepts that are basic to the releaser process:

- Age
- Candidate
- Priority

- Weight
- Partial Release

Age

Age is the amount of time that elapsed from a given event to the present. A file's inode keeps track of the following times:

- Residence-change time
- Data-modified time
- Data-accessed time

You can view these times by using the `sls`(1) command with the `-D` option. Each time has a corresponding age. For example, if it is 10:15 a.m., a file with a modify time of 10:10 a.m. has a data-modified age of five minutes. For more information, see the `sls`(1) man page.

Candidate

A candidate is a file that is eligible to be released. A file is not a candidate under the following circumstances:

- The file is already offline.
- The file has not been archived.
- The `archiver.cmd` command file specifies the `-norelease` attribute for the file and the required copies have not yet been made.
- The file is marked as damaged.
- The file is not a directory, block, character-special file, or pipe.
- The archiver is staging the file to make an additional copy. The file becomes eligible for release after the archiver stages it.
- The age of the file is negative. This condition occurs for network file system (NFS) clients with inaccurate clock settings.
- The file is marked to never be released. You can use the `release -n` command to specify this.
- The file was staged at a time in the past that is less than the minimum residence time setting. For more information, see The min_residence_age Directive: Specifying a Minimum Residence Time.
- The file was flagged for partial release, through the `release`(1) command's `-p` option, and it is already partially released.
- The file is too small. Releasing it will not create much space.

Priority

A priority is a numeric value that indicates the rank of a candidate file based on user-supplied weights that are applied to numeric attributes of that candidate. The overall priority is the sum of two types of priority: age priority and size priority. Candidate files with numerically larger priorities are released before candidates with numerically smaller priorities.

Weight

A weight is a numeric value that biases the priority calculation to include file attributes in which you are interested and to exclude file attributes in which you are not interested. For example, if the size weight is set to 0, the size attribute of a file is excluded from the priority calculation. Weights are floating-point values from 0.0 to 1.0.

Partial Release

With partial release, a beginning portion of the file remains in disk cache while the rest of the file is released. Partial release is useful with utilities such as `filemgr`(1) that read the beginning of a file.

About Partial Releasing and Partial Staging

Releasing and staging are complementary processes. Files can be completely released from online disk cache after they are archived, or a site can specify that the beginning of a file (the stub) remain in disk cache while the remainder of the file is released. Partially releasing a file provides immediate access to data in the file stub without requiring that the file be staged. You can specify both the default partial release size and the maximum size of the stub to remain online when a file system is mounted.

You can set the through the values for partial releasing and staging using the `mount`(1M) command or in the SAM-QFS Manager. See the SAM-QFS Manager online help for more information.

The `mount`(1M) command options are as follows:

- `-o partial=n` option - Sets the default size (n) of a file stub to remain online. The `-o partial=n` setting must be less than or equal to the `-o maxpartial=n` setting.
- `-o maxpartial=n` option - Sets the maximum size (n) of a file stub to remain online.

You can specify the default stub size for a file by specifying the `-p` option on the `release`(1) command or the p option on the `sam_release`(3) library routine. To specify different-sized file stubs for different types of files or different applications, specify the `-s` option on the `release`(1) command or the s option on the `sam_release`(3) library routine. The `-s` and s values must be less than the `-o maxpartial` value used with the `mount`(1M) command when the file system was mounted.

> Note
> A partially released file takes up space on the disk equal to one DAU. For example, if the partial release file stub is set to 16K and the DAU size is 256K, the actual space consumed by the file on the disk is 256K.

Use the mount option, `-o partial_stage=n`, to establish how much of a partial release stub must be read before the rest of the file is staged. Reading past the `-o partial_stage=n` size specification initiates the stage of the file.
By default, the `-o partial_stage=n` option is set to the size of the partial release stub. Changing this value affects file staging as follows:

- If the `-o partial_stage=n` option is set to the size of the partial release stub, the default behavior prevents the file from being staged until the application reaches the end of the partial release stub. Waiting until the end of the stub is reached causes a delay in accessing the rest of the file.
- If the `-o partial_stage=n` option is set to a value smaller than the partial release stub, the file is staged after the application crosses the threshold set by the `-o partial_stage=n` option. This reduces the chance of a delay in accessing the rest of the file data.

Example - Partial Staging

In this example, a site has set the following options:

- `-o partial_stage=16` (16 kilobytes)
- `-o partial=2097152` (2 gigabytes)
- `-o maxpartial=2097152` (2 gigabytes)

The `filemgr`(1) program reads the first 8 kilobytes of a file. The file is not staged.
A video-on-demand application reads the same file. After it reads past the first 16 kilobytes of the file, the file is staged. The application continues reading while the archive tape is mounted and positioned.
When the video-on-demand application reads past two gigabytes of file data, it is reading immediately behind the staging activity. The application does not wait, because the tape mounting and positioning is done while the application reads the partial file data.

Several command-line options affect whether a file can be marked for partial release. Some options are enabled by the system administrator, and others can be enabled by individual users. The following sections describe the release characteristics that can be set by the various types of users.

- System Administrator Option Summary
- User Option Summary

Summary of System Administrator Options

As a system administrator, you can change the maximum value and default value for partial release when the file system is mounted. The mount(1M) options in the following table affect partial release. For more information about the mount(1) command, see the mount_samfs(1M) man page.

Table - Mount Options for Partial Release

Option	Effect
`-o maxpartial=n`	Determines the maximum amount of space in kilobytes that can remain in disk cache when a file is marked for partial release. The maximum value is 2,097,152 kilobytes, which is 2 gigabytes. The minimum value is 0, which disables the partial release feature is disabled so that released files are released completely, and no portion of a file remains in disk cache. Users cannot override the value specified on this option after the file system is mounted. By default, the n argument is set to 16.
`-o partial=n`	Sets a default amount of space in kilobytes that remains in disk cache when a user marks a file for partial release by using the `release`(1) command's `-p` option. The n argument must be at least 8, but it can be as great as the value specified for the `-o maxpartial=n` option. Because some applications do not need access to the entire file to complete their operations, this option ensures that applications have the beginnings of files available to them. Also, this option prevents files from being staged unnecessarily. By default, n is 16. A file that has been partially released from a disk takes up space on the disk equal to one DAU.

-o partial_stage=n	Specifies that when a partially released file is accessed, n bytes of the file must be read before the entire file is staged from the archive media. Set this value lower than the amount of the -o partial setting. For n, specify an integer from 0 to the -o maxpartial specification. By default, this is 16 or the value specified for the -o partial option.
-o stage_n_window=n	Specifies the amount of data to be staged at any one time to n. Specify an integer from 64 to 2,048,000. The default is 256 kilobytes. This option applies only to files that have the stage -n attribute set.

Summary of User Options

As a user, you can set maximum and default values for the size of a file stub that can remain in disk cache after the file is released. You can also determine whether the partial release feature is enabled for a particular file system.

By using the release(1) command and the sam_release(3) library routines, however, a user can set other release attributes and can specify the files to be marked for partial release. The command and library options that determine partial release attributes are shown in the following table. For more information, see the release(1) man page and thee sam_release(3) man page.

Table - User Release Options

Options	Effect
release(1) command and -p option or sam_release(3) library routine and p option	The -p and p options mark the named file for partial release. If these options are used, the amount of the file remaining in online disk cache after the file is released depends on the value of the -o partial=n option that was set when the file system in which the file resides was mounted. These options cannot be used to specify the number of bytes to remain online.
release(1) command and -s partial_size option or sam_release(3) library routine and s option	The -s and s options mark the named file for partial release, and they specify the amount of the file to remain in online disk cache. The arguments to the -s or s options specify the amount, in kilobytes, to remain online. A user cannot specify that the amount of a file remaining online be greater than the amount specified for the -o maxpartial=n value when the file system was mounted. If the user's value is greater than the value for the file system, the value for the file system is used, and the user's specification is ignored.

About Staging

Staging is the process of copying file data from nearline or offline storage back to online storage.

The stager starts when the samd daemon runs. The stager has the following default behavior:

- The stager attempts to use all the drives in the library.
- The stage buffer size is determined by the media type, and the stage buffer is not locked.
- No log file is written.
- Up to 1000 stage requests can be active at any one time.
  You can customize the stager's operations for your site by inserting directives into the /etc/opt/SUNWsamfs/stager.cmd file.

When an application requires an offline file, its archive copy is staged to disk cache (if the -n option's -stage never is not set). To make the file available to an application immediately, the read operation tracks along directly behind the staging operation so that the access can begin before the entire file is staged.

Stage errors that include media errors, unavailability of media, unavailability of an automated library, and others. If a stage error is returned, the SAM-QFS software attempts to find the next available copy of the file, if one exists and if there is a device available to read the archive copy's media.

See Configuring the Stager for details on how to configure the stager through the stager.cmd file.

Configuring the Stager

This page shows how to configure the stager through the stager.cmd file.

About the stager.cmd File

In the the stager.cmd file, specify directives to override the default behaviors. You can configure the stager to stage files immediately, to never stage files, to staging partially, and to specify other staging actions. For example, specifying the never-stage attribute benefits applications that access small records from large files because the data is accessed directly from the archive media without staging the file online.

This section describes the stager directives. For additional information about stager directives, see the stager.cmd(4) man page. If you are using the SAM-QFS Manager software, you can control staging from the File System Summary or File System Details page. You can browse the file system and see the status of individual files, use filters to view certain files, and select specific files to stage. You can select which copy to stage from or let the system choose the copy.

Example ***stager.cmd*** File

The following example shows a stager.cmd file after all possible directives have been set.

```
# This is stager.cmd file /etc/opt/SUNWsamfs/stager.cmd
drives=dog 1
bufsize=od 8 lock
logfile=/var/adm/stage.log
maxactive=500
```

How to Create a stager.cmd File

1. In the /etc/opt/SUNWsamfs/stager.cmd file, add the directives to control staging at your site, according to the information in the following sections:
   - drives Directive: Specifying the Number of Drives for Staging
   - bufsize Directive: Setting the Stage Buffer Size
   - logfile Directive: Specifying a Log File
   - maxactive Directive: Specifying the Number of Stage Requests
2. Save and close the stager.cmd file.
3. Propagate the file changes and restart the system.

```
# samd config
```

drives Directive: Specifying the Number of Drives for Staging

By default, the stager uses all available drives when staging files. If the stager keeps all the drives busy, it can interfere with the archiver's activities. The drives directive specifies the number of drives available to the stager. This directive has the following format:

```
drives = <library> <count>
```

Argument	Meaning
library	The family set name of a library as it appears in the mcf file.
count	The maximum number of drives to be used. By default, this is the number of drives configured in the mcf file for this library.

You can also specify this directive by using the SAM-QFS Manager software. For more information, see the SAM-QFS Manager online help.

Example - Drives Directive

The following example specifies that only one drive from the dog family set's library is used for staging files:

```
drives = dog 1
```

bufsize Directive: Setting the Stage Buffer Size

By default, a file being staged is read into memory in a buffer before being restored from the archive media to disk cache. Use the `bufsize` directive to specify a buffer size and, optionally, to lock the buffer. These actions can improve performance. You can experiment with various buffer-size values. The directive has the following format:

```
bufsize = <media> <buffer-size> [lock]
```

Argument	Meaning
media	Specify the archive media type from the list on the `mcf`(4) man page.
buffer-size	A number from 2 through 8192. The default is 16. This value is multiplied by the dev*`blksize`* value for the media type, and the resulting buffer size is used. The dev*`blksize`* value is specified in the `defaults.conf` file. The higher the number specified for buffer-size, the more memory is used. For more information, see the `defaults.conf`(4) man page.
`lock`	The `lock` argument indicates that the stager should use locked buffers when staging archive copies. If `lock` is specified, the stager sets file locks on the stage buffer in memory for the duration of the copy operation. This avoids the overhead associated with locking and unlocking the buffer for each I/O request and can thereby result in a reduction in system CPU time.The `lock` argument should be specified only on large systems with large amounts of memory. Insufficient memory can cause an out-of-memory condition.The `lock` argument is effective only if direct I/O is enabled for the file being staged. By default, `lock` is not specified, and the file system sets the locks on all direct I/O buffers, including those for staging. For more information about enabling direct I/O, see the `setfa`(1) man page, the `sam_setfa`(3) library routine man page, or the `-O forcedirectio` option on the `mount_samfs`(1M) man page.

You can also specify this directive by using the SAM-QFS Manager software. For more information, see the SAM-QFS Manager online help.

logfile Directive: Specifying a Log File

You can request that the SAM-QFS software collect file-staging event information and write it to a log file. By default, no log file is written. The `logfile` directive specifies a log file to which the stager can write logging information. The stager writes one or more lines to the log file for each file staged. This line includes information such as the name of the file, the date and time of the stage, and the volume serial number (VSN). The directive has the following format:

```
logfile=<filename> [<event>]
```

Argument	Meaning
filename	Specify a full path name.
event	Specify one or more staging events. If you specify more than one event, use spaces to separate each them. Possible event specifications are as follows. `all` - Logs all staging events. `start` - Logs when staging begins for a file. `finish` - Logs when staging ends for a file. Enabled by default. `cancel` - Logs when the operator cancels a stage. Enabled by default. `error` - Logs staging errors. Enabled by default.

You can also specify this directive by using the SAM-QFS Manager software. For more information, see the SAM-QFS Manager online help.

Example: Specifying a Stager Log File

The following directive creates the `/var/adm/stage.log` file:

```
logfile=/var/adm/stage.log
```

Example: Stager Log File

```
S 2003/12/16 14:06:27 dk disk01 e.76d 2557.1759 1743132 /sam1/testdir0/filebu 1 root other root 0
-
F 2003/12/16 14:06:27 dk disk01 e.76d 2557.1759 1743132 /sam1/testdir0/filebu 1 root other root 0
-
S 2003/12/16 14:06:27 dk disk02 4.a68 1218.1387 519464 /sam1/testdir1/fileaq 1 root other root 0 -
S 2003/12/16 14:06:43 dk disk01 13.ba5 3179.41 750880 /sam1/testdir0/filecl 1 root other root 0 -
F 2003/12/16 14:06:43 dk disk01 13.ba5 3179.41 750880 /sam1/testdir0/filecl 1 root other root 0 -
S 2003/12/16 14:06:59 dk disk01 17.167b 1155.1677 1354160 /sam1/testdir0/filedb 1 root other root
0 -
F 2003/12/16 14:06:59 dk disk01 17.167b 1155.1677 1354160 /sam1/testdir0/filedb 1 root other root
0 -
S 2003/12/16 14:06:59 dk disk02 f.f82 3501.115 1458848 /sam1/testdir1/filecb 1 root other root 0 -
S 2003/12/16 14:07:15 dk disk01 1f.473 1368.1419 636473 /sam1/testdir0/fileed 1 root other root 0
-
S 2003/12/16 14:07:15 dk disk02 16.f15 3362.45 1065457 /sam1/testdir1/filecz 1 root other root 0 -
S 2003/12/16 14:07:31 dk disk01 23.201d 3005.1381 556807 /sam1/testdir0/fileeq 1 root other root 0
-
S 2003/12/16 14:07:47 dk disk01 26.c4d 2831.1113 1428718 /sam1/testdir0/fileez 1 root other root 0
-
S 2003/12/16 14:07:47 dk disk02 1b.835 3736.59 1787855 /sam1/testdir1/filedp 1 root other root 0 -
```

The following describes the content of the stager log file fields.

Field	Example Value	Content Description
1	S	Stage activity - S for start, C for canceled, E for error, F for finished.
2	2003/12/16	Date of the stage action, in yyyy/mm/dd format.
3	14:06:27	Time of the stage action, in hh:mm:ss format.
4	dk	Archive media type. For information about media types, see the `mcf`(4) man page.
5	disk01	VSN.
6	e.76d	Using hexadecimal format, the physical position of the start of the archive file on media (`tar`(1) file) and the file offset on the archive file.
7	2557.1759	Inode number and generation number. The generation number is used in addition to the inode number for uniqueness because inode numbers are reused.
8	1743132	Length of the file.
9	`/sam1/testdir0/filebu`	Name of the file.
10	1	Archive copy number.
11	root	User ID of the file.
12	other	Group ID of the file.
13	root	Group ID of the requestor.
14	0	Equipment ordinal of the drive from which the file was staged.
15	-	A V in this field indicates that data verification is being used for the file.

maxactive Directive: Specifying the Number of Stage Requests

The `maxactive` directive enables you to specify the number of stage requests that can be active at any one time. The directive has the following format:

```
maxactive=<number>
```

By default, number is 4000. The minimum number allowed is 1. The maximum allowed is 500,000.

Example - Number of Stage Requests

The following example specifies that no more than 500 stage requests can be in the queue simultaneously:

```
maxactive=500
```

Archive Set Assignment Directive: Specifying Stage Attributes for All Files in an Archive Set

Most directives in the `archiver.cmd` file affect only archiving, but you can use the archive set assignment directive to specify stage attributes that apply to all files in an archive set.

The Archive Set Directives (archiver.cmd) page describes the archive set assignment directive and its arguments completely. The following table shows the staging directives that can appear in an archive set assignment directive in the `archiver.cmd` file.

Directive	Effect
`-stage a`	Specifies that the files in the archive set should be associatively staged.
`-stage d`	Reset to default.
`-stage n`	Specifies that the files in the archive set should never be staged.

Prioritizing Preview Requests

Both the archiver and stager processes request that media is loaded and unloaded. If the number of requests exceeds the number of drives available for media loads, the excess requests are sent to the preview queue.

The number of entries that can be in the preview queue is determined by the `previews=` directive in the `defaults.conf` file. For information about changing the value of this directive, see the `defaults.conf`(4) man page.

By default, preview requests are satisfied in first-in-first-out (FIFO) order.
The overall priority of preview requests is determined by the combination of static and dynamic factors. Higher numbers correspond to higher priority. A static priority factor is set when the request is generated. Its effect does not change the overall priority after the request is generated and is waiting to be satisfied. A dynamic priority factor can increase or decrease the overall priority of a request while the request is waiting to be satisfied.

You can override the FIFO default by entering directives in the `/etc/opt/SUNWsamfs/preview.cmd` command file.

About the preview.cmd File

The `sam-amld` daemon reads the `preview.cmd` file at startup. This file orders the requests in the preview queue according to whether the request is for staging or archiving. You can increase the priority for specific VSNs and you can control the priority of preview requests for specific file systems.

The following rules apply to the `preview.cmd` file:

- Place one directive per line.
- If you change this file while the `sam-amld` daemon is running, restart the daemon to have your changes take effect.
- Begin comment lines begin with a pound sign (#).
  For more information about this file, see the `preview.cmd`(4) man page.

The following types of directives are used in the `preview.cmd` file:

- Global directives, which apply to all file systems
- File-system directives, specific to individual file systems
  Global directives are placed at the top of the file and their settings apply to all file systems.

File system directives begin with the `fs =` directive, which names the file system to which all subsequent directives apply. More than one block of file directives can appear in a file. File system directives apply until the next `fs =` line is encountered or until the end of file is encountered.

When multiple directives affect a file system, the directives that are specific to that file system override the global directives.

In the following sections, edit the `preview.cmd` file to control the preview queue, according to the information in the following section:

- How to Set the Global VSN and Age Priority Directives
- How to Set Global or File-System-Specific Water Mark Directives
- Setting Up a Preview Request Priority Scheme

How to Set the Global VSN and Age Priority Directives

> Info -
> The VSN and age priority directives are global directives, so they are placed before any file-system-specific directives in the `preview.cmd` file.

1. Update the `vsn_priority` directive.

```
vsn_priority = <value>
```

   This directive is a static priority factor that indicates the value by which the total priority increases when there is a high-priority volume. The default value for is `1000.0`. To use this priority factor, a volume must have its priority flag set before it is scheduled as a preview request. Use the `chmed`(1M) command to set the priority flag with the p option (for example, `chmed +p lt.`AAA123).

2. Update the `age_priority` directive.

```
age_priority = <factor>
```

   This directive is a static priority factor, although its effect is dynamic. This factor is multiplied by the number of seconds for which a request is a preview request. The result is then added to the overall priority of the request. The longer a request waits, the higher the priority becomes. Setting this factor ensures that older requests are not indefinitely superseded by newer requests with other higher-priority factors.
   Setting this factor to more than 1.0 increases the importance of the time factor in calculation of the total priority and setting it to less than 1.0 decreases the importance of the time factor. Setting the factor to 0.0 eliminates the time factor from the overall priority calculation.
   A volume whose priority flag is not set increases in priority based on the time it remains in the queue. Its priority can become higher than a VSN that comes into the queue later with the priority flag already set.

How to Set Global or File-System-Specific Water Mark Directives

The water mark preview request directives can be used as either global or file-system-specific directives. These directives determine the water mark priority of the preview requests, as shown in the following equation.

```
lwm_priority + lhwm_priority + hlwm_priority + hwm_priority = water mark priority
```

Together, the four water mark settings create a dynamic priority factor that includes a percentage value indicating how full the file system is and the levels at which the HWM and LWM are set. The value assigned to a preview request is determined by whether a factor is global, specific to a file system, or not set.

The water mark priorities are used to calculate only requests for archiving. They are not used to calculate media requests for staging. When the water mark priority factor is a positive number, the result on the overall calculated priorities is to raise archiving requests over staging requests. In contrast, when the water mark priority factor is a negative number, the overall priority for archiving requests is reduced, which tends to favor staging requests over archival requests. A water mark priority factor of 0.0 (or no specified directive) indicates that no special action occurs.

For more information, see Example 1 - Scheme for Enforcing Stage Requests.

The water mark directives have the following format:

```
<wmtype>_priority = <value>
```

Water Mark Directive	Argument
lwm_priority = value	Specify the amount by which you want the water mark priority factor to change for archiving requests when the file system is below the LWM level. The default is 0.0.
lhwm_priority = value	Specify the amount by which you want the water mark priority factor to change for archiving requests when the file system crosses from below to above the LWM level but remains below the HWM level. This shift indicates that the file system is filling up. The default is 0.0.
hlwm_priority = value	Specify the amount by which you want the water mark priority factor to change for archiving requests when the file system has crossed from above the HWM level to below it, but remains above the LWM level. This shift indicates that the releaser was not able to free enough disk space to leave the file system below the LWM level. The default is 0.0.
hwm_priority = value	Specify the amount by which you want the water mark priority factor to change for archiving requests when the file system is above the HWM level. The default is 0.0.

When a file system crosses from one condition to another, the priority of each volume associated with that file system is recalculated based on the appropriate water mark priority setting, with or without the chmed(1M) command's p option.

Example: Settings for a File System Below the LWM Level

The following example frees enough disk space so that the file system goes below the LWM level.

```
lhwm_priority = -200.0
hlwm_priority = 100.0
```

Setting Up a Preview Request Priority Scheme

The total priority for a preview request is the sum of all priority factors:

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

Change the default FIFO scheme only for reasons, such as the following:

- Ensure that staging requests are processed before archive requests.
- Ensure that archive requests gain top priority when a file system is about to fill up.
- Push requests that use a specific group of media to the top of the preview request list.
  The following example shows a preview.cmd file that addresses these three conditions.

Example - Sample ***preview.cmd*** File

```
# 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
```

For environments in which user access to data is of paramount importance, the VSN drives are limited, or file archiving is performed as a background function, use the preview.cmd file to influence how the storage system resources handle staging requests. You can customize the settings in the preview.cmd file to support any of the preceding scenarios and influence the configured SAM-QFS environment.

Because data is not affected by the settings in this file, you are encouraged to experiment and adjust the directive settings to achieve the proper balance between archiving and staging requests when weighed against the priorities of each preview request.

Example 1 - Scheme for Enforcing Stage Requests

The following example calculations show how you can use a negative value for `wm_priority` to ensure that stage requests have priority over archive requests. This example assumes the following:

- Several requests are sitting in the queue for 100 seconds.
- The default value `vsn_priority` is 1000.

The following table shows how the total request priorities are calculated.

Priority	Calculation
Archive VSN with priority, LWM:	1000 + (-200) + (1 x 100) = 900
Stage VSN with priority, LWM:	1000 + 0 + (1 x 100) = 1100
Stage VSN without priority, LWM:	0 + 0 + (1 x 100) = 100

Example 2 - Scheme for Enforcing Archive Requests

When the environment is balanced between the importance of staging a file for the user and the importance of getting new files archived to media, the biggest concern is exceeding the HWM level. In this situation, if not enough files have met their archive requirements to lower the percentage of the file system that is full, meeting the pending archive requests is the best way to keep the file system from reaching its limit. In this situation, the `preview.cmd` file can be as simple as the following:

```
hwm_priority = 500.0
```

Example 3 - Scheme for Ranking Requests by Media Type

A site has an environment in which users are working on groups of files that use specific volumes and are segregated from other users. In this environment, certain projects might have higher priorities at certain times; therefore, greater priority is required from the available system storage resources. The following example gives users and their media the appropriate priority:

```
hwm_priority = 5000.0
```

Then, for every volume in the priority user's group, include the following information:

```
# chmed +p lt. <VSN>
```

Now every request that requires the specified VSN is placed above other pending mount requests in the preview queue.
Later, to lower the priority of the user's media, include the following reverse command for every VSN:

```
# chmed -p lt. <media-type>
```

> Note -
> A request for a select group of VSNs always takes precedence in the preview request queue if the `chmed(1M)` command's p flag is set.

Example 4 - Scheme for Complex Priorities

Assume that there are two file systems with the following requirements:

- No request must wait too long in the queue (`age_priority`).
- When one of the file systems is below the LWM level, staging requests take precedence.
- When one of the file systems is above the LWM level but below the HWM level, do not prioritize requests.

The following example shows the affected directives.

```
lwm_priority = -200.0
lhwm_priority = 0.0
hlwm_priority = 0.0
```

When one of the file systems goes over the HWM level, archive requests take priority.

Assume both file systems are over the HWM level but it is important to prevent the second file system (`samfs2`) from reaching its limit. The following example shows a `preview.cmd` file that prioritizes requests according to the requirements in the preceding list.

```
age_priority = 100.0
vsn_priority = 20000.0
lhwm_priority = -200.0
hlwm_priority = -200.0
fs = samfs1
hwm_priority = 1000.0
fs = samfs2
hwm_priority = 5000.0
```

About Recycling

Recycling is the process of reclaiming space on archive volumes. The recycler works with the archiver to reclaim the space occupied by unused archive copies. As users modify files, the archive copies associated with the old versions can be purged from the system. The recycler identifies the volumes with the largest proportions of expired archive copies and directs the movement of unexpired copies to different volumes. If only expired copies exist on a given volume, a site-defined action is taken. For example, a volume can be relabeled for immediate reuse or exported to offsite storage, keeping a separate historical record of file changes. Users are unaware of the recycling process.

At any time, the space on an archive volume consists of the following:

- Current data, consisting of archive images that are active
- Expired data, consisting of archive images that are no longer active
- Free space, consisting of space that is not being used by active or expired archive images

The recycler keeps the amount of space consumed by expired data to the minimum defined by site-specified parameters.

The capacity of a volume is the total amount of space for data on a volume. For example, a 10-gigabyte volume with 3 gigabytes written to it has a capacity of 10 gigabytes and 7 gigabytes of free space.
New or newly-labeled archive media starts with all its capacity as free space. As data is archived to the media, the amount of free space decreases and the amount of current data increases.

As files in the file system are changed or removed, their archive images expire and the classification of their data changes from the current data to the expired data. The physical space used by the archive images does not change. However, no file in the file system points to that space. When space is recycled, these images are removed and the space they occupied become free, available for other purposes. The goal of the recycler is to transform space used by expired data into free space without losing any current data.

The Recycle Process

The recycler and the archiver work together, as follows:

1. The recycler marks all the current archive images that are present on a volume with the `rearchive` attribute.
2. If you are archiving to removable media, the recycler marks the archive volume with the `recycle` attribute to prevent the archiver from writing any more archive images to the volume.
3. The archiver moves all the marked images to another volume. This operation is called re-archiving. After the archiver moves the current archive images from the old volume to the new volume, the old volume contains only free space and expired space. If you are archiving to removable media cartridges, you can relabel and reuse the cartridge. If you are archiving to disk, the recycler removes the file that contains the expired archive images.

The recycler is designed to run periodically, although you can run it at any time. It performs as much work as it can each time it is invoked. The recycler has to finish marking copies for re-archiving before the archiver can re-archive the files. Sometimes expired archive images, with the `rearchive` attribute set, remain on media. This can happen under the following conditions:

- The archiver does not run after the recycler marks expired archive images.
- Media is not available for the archiver to use when moving the unexpired archive images.
- Miscellaneous archiver anomalies occur.

Between executions, the recycler keeps state information in the library catalogs and the inodes. During the recycling process, you can use the `sls`(1) command and its `-D` option to display information about a file. The output from the `sls`(1) command shows whether a file is scheduled for re-archiving.

Planning for Recycling

Recycling is accomplished using two methods, depending on the type of media.

Media Types and Recycling Methods

Archive Media	Recycling Method
Removable media cartridges	By automated library
Removable media cartridges	By archive set
Disks	By archive set

For information about configuring by these methods, see Configuring the Recycler.

Configuring the Recycler

This page describes the recycling process and directives.

Recycling Methods

You initiate recycling by entering the `sam-recycler`(1M) command either manually or through a `cron`(1) job. The following table shows the recycling methods.

Recycling Methods and Media Types

Recycling Method	Archive Media	Where To Configure
By automated library	Removable media cartridges	`recycler.cmd` `recycler.sh`

By archive set	Removable media cartridges	`recycler.cmd` (optional, for VSNs not covered in archive set)
		`recycler.sh`
		`archiver.cmd`

By archive set	Disks	`archiver.cmd`

Before configuring the recycler, note the following:

- Directives in the `archiver.cmd` file control recycling by archive set.
  Directives in the `recycler.cmd` file control recycling by library. In addition, the `recycler.cmd` file controls general recycler behavior. For information about recycler directives, see Creating a `recycler.cmd` File.
- Do not recycle volumes that contain removable media files. You create removable media files by using the `request`(1) command. A volume with removable media files can never be drained.
- Do not run the recycler while performing maintenance on a file system. The recycler uses the `.inodes` file and the `mcf` file to determine whether files are current or expired and to identify the devices associated with a file system. Absence of proper information in the `.inodes` and `mcf` files can cause current archived data to appear as expired and be recycled.
- All file systems must be mounted when the recycler is run. If you are recycling from online disk, the file system that contains the disk volumes must be mounted and the host system must be accessible.

Caution
Take extreme care when configuring the recycler if you are using disk archiving in an environment with multiple SAM-QFS servers. The `diskvols.conf` file for each SAM-QFS server must point to a unique set of disk archiving target directories. If any of these directories are shared by different SAM-QFS servers, running the recycler from one SAM-QFS server will destroy the disk archive data that is managed by the other SAM-QFS server.

Controlling Recycling

You can enable and disable the recycle process using parameters in the command files.
When you are configuring the recycler and testing the results, edit the appropriate file to add its parameter.

If You Recycle By...	Add The Parameter
Archive Set	`-recycle_ignore` parameter to the `archiver.cmd` file.
Automated Library	`ignore` parameter to the `recycler.cmd` files.

When you are ready to use the recycler, edit the files to remove the parameter.
As an alternative, you can use the SAM-QFS Manager. For more information, see the SAM-QFS Manager online help.

How to Start the Recycler

Follow these instructions to run the recycler.

1. Run the `sam-recycler`(1M) command.

```
# sam-recycler
```

   The recycler reads the `recycler.cmd` file.

2. Examine the standard output log, SAM-QFS log, and `/var/adm/messages` for any error messages from the recycler.
   Correct your files if errors appear.

When the recycler is initiated, the default recycler settings specified in library Directive: Specifying Recycling for an Entire Automated Library take effect. For more information on the recycler, see the `sam-recycler`(1M) man page.

How to Run the Recycler Periodically

If the system is performing in a routine manner, you can make a `crontab` entry. The frequency you choose depends on your site's conditions.
For instructions about creating a `crontab` entry, see the `cron`(1M) man page.

The following example entry in root's `crontab` file specifies that the `cron` daemon run the recycler every five minutes after every odd-numbered hour:

```
5 1,3,5,7,9,11,13,15,17,19,21,23 * * * /opt/SUNWsamfs/sbin/sam-recycler
```

Configuring Recycling on Removable Media Cartridges

If you are recycling archive copies on cartridges in a library, create a `recycler.cmd` file.
If you are recycling by archive set, configure each library in the `recycler.cmd` file. This ensures that VSNs that do not fall into an archive set and can be recycled if needed.
Create a `recycler.sh` file to complete the operation.

Creating a **`recycler.cmd`** File

The `recycler.cmd` file contains general recycling directives. It can also contain directives for each library in the SAM-QFS environment. A typical recycler.cmd file contains the following directive lines:

- A logfile= directive to specify a recycler log file.
- One or more directives for each library that contains volumes to be recycled. This line must contain the family set name (from the mcf file) for the library being recycled. The family set name identifies the library to the recycler. For information, see Using Recycling Directives.
- During testing, include the ignore keyword. You remove the ignore keyword in a later step in this process.

1. Become superuser.
2. In the /etc/opt/SUNWsamfs/recycler.cmd file, add one or more directives to control recycler activity.
3. Save and close the file.
   Aa an alternative, you can create a recycler.cmd file using SAM-QFS Manager software. For more information, see the SAM-QFS Manager online help.

Example - Sample *recycler.cmd* File

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

The recycler.cmd file accepts the directives described in the following sections:

- logfile Directive: Specifying a Log File
- no_recycle Directive: Preventing Recycling
- library Directive: Specifying Recycling for an Entire Automated Library

logfile Directive: Specifying a Log File

The logfile directive creates a recycler log file. This directive has the following format:

```
logfile = <filename>
```

For filename, specify the path to the log file.
The following is an example of a logfile= directive line:

```
logfile=/var/adm/recycler.log
```

Example - Sample Recycler Log File for Removable Media Cartridges

The following example shows a sample recycler log file for recycling removable media cartridges.

```
========== Recycler begins at Wed Dec 12 14:05:21 2001 ===========
Initial 2 catalogs:
0 Family: m160 Path: /var/opt/SUNWsamfs/catalog/m160
Vendor: ADIC Product: Scalar 100
SLOT ty capacity space vsn
0 at 25.0G 25.0G CLN005
1 at 48.5G 6.1G 000003
2 at 48.5G 32.1G 000004
3 at 48.5G 35.1G 000005
4 at 48.5G 44.6G 000044
5 at 48.5G 45.1G 000002
6 at 48.5G 45.9G 000033
7 at 48.5G 48.5G 000001
Total Capacity: 364.8G bytes, Total Space Available: 282.3G bytes
Volume utilization 22%, high 95% VSN_min 50%
Recycling is ignored on this robot.
1 Family: hy Path: /var/opt/SUNWsamfs/catalog/historian
Vendor: Sun SAM-FS Product: Historian
SLOT ty capacity space vsn
(no VSNs in this media changer)
Total Capacity: 0 bytes, Total Space Available: 0 bytes
Volume utilization 0%, high 95% VSN_min 50%
Recycling is ignored on this robot.
8 VSNs:
---Archives--- -----Percent----- m160
----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 87 13 m160:at:000003
no-data VSN 0 0 0 33 67 m160:at:000004
no-data VSN 0 0 0 27 73 m160:at:000005
no-data VSN 0 0 0 8 92 m160:at:000044
no-data VSN 0 0 0 7 93 m160:at:000002
no-data VSN 0 0 0 5 95 m160:at:000033
empty VSN 0 0 0 0 100 m160:at:CLN005
empty VSN 0 0 0 0 100 m160:at:000001
Recycler finished.
========== Recycler ends at Wed Dec 12 14:05:32 2001 ===========
```

no_recycle Directive: Preventing Recycling

The `no_recycle` directive disables recycling of volumes. This directive has the following format:

```
no_recycle <media-type> <VSN-regexP> [<VSN-regexP>...]
```

Argument	Meaning
media-type	A media type from the `mcf`(4) man page. You can disable recycling of volumes stored on particular type of media
VSN-regexp	One or more space-separated regular expressions to describe the volumes. You can disable recycling for specific cartridges. For information, see the `regexp`(5) man page or see File Name search-criterion Using Pattern Matching: `-name regex`.

Example - Preventing Recycling of Specific Cartridges

The following example excludes any tape volumes whose VSN identifiers begin with `DLT`:

```
no_recycle lt DLT.*
```

library Directive: Specifying Recycling for an Automated Library

The library directive enables you to specify various recycling parameters for the VSNs associated with a specific library. This directive has the following format:

```
<library> <parameter> [<parameter>...]
```

For library, specify the library's name as specified in the family set field of the `mcf` file.
For parameter, specify one or more space-separated parameter keywords from the following table.

Parameter Value	Meaning
`-dataquantity` size	Maximum amount of data that the recycler can schedule for rearchiving in its efforts to clear volumes of useful data. Default is 1 gigabyte.
`-hwm` percent	Library high-water mark. Default is 95.
`-ignore`	Directive that prevents volumes in this library from being recycled. This directive is useful during testing of the `recycler.cmd` file.
`-mail` email-address	Email addresses to which recycling email messages are to be sent. By default, no email is sent.
`-mingain` value	Minimum VSN gain. Default depends on media: • For volumes with less than 200 GByte capacity, the default mingain is 60%. • For volumes with 200 GByte or larger capacity, the default mingain is 90%.
`-vsncount` count	Maximum number of recycled volumes to be counted. Default is 1.

Example - library Directive

The following example specifies the following for library `gr47`:

- The library qualifies for recycling when the volumes in the library are 85 percent full.
- The minimum percent gain is 40 percent.
- Only one volume is to be recycled. This is also a default setting.
- Recycling messages are emailed to `root`.
- No more than 1 gigabyte is to be re-archived. This is the default, so it is not specified in the `recycler.cmd` file.

```
gr47 -hwm 85 -ignore -mail root -mingain 40
```

The following sections describe the parameters.

-hwm Parameter

By specifying a high-water mark, you set the percentage of media usage below which recycling cannot occur. This percentage is the ratio of the used space in the library to its total capacity. For example, a library that holds ten 20-gigabyte tapes, three of them 100 percent full and the remaining seven each 30 percent full, has the following media utilization percentage:

```
((3* 1.00 + 7 * 0.30) * 20G ) / ( 10 * 20G ) * 100%= 51%
```

This calculation does not distinguish between current data and expired data. It only addresses the amount of media used.

In the example, when the utilization percentage is 51 percent or less, the recycler does not automatically select any of the automated library's VSNs for recycling.

You can force a VSN to be recycled by using the following command to set the recycling flag:

```
# chmed +c lt. <VSN>
```

When the `+c` flag is set, the archiver does not write any more archive images to the volume. The `+c` flag can be viewed through the `samu`(1M) utility. For more information, see the `chmed`(1M) and `samu`(1M) man pages. For information about using the `samu`(1M) operator utility, see the

Sun QFS File System Configuration and Administration Guide.

-mingain Parameter

The minimum VSN gain percentage sets a lower limit on the amount of space to be gained by recycling a cartridge. For example, if a cartridge in an automated library is 95 percent current data and 5 percent is efficient. Setting the minimum gain to 6 percent or more inhibits the recycler from automatically selecting this VSN.

-ignore Parameter

The `-ignore` parameter disables the recycler for a particular library. Use it while you are configuring and testing the recycler.

-mail Parameter

The `-mail` parameter specifies that the recycler sends email messages when recycling occurs on a library. The email message has the following subject line:

```
Robot <robot-name> recycle
```

Example - Sample Recycling Messages

```
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.
```

Creating a **recycler.sh** File

If you are archiving on removable media cartridges, create a `recycler.sh`.
If you are archiving only to disk, do not perform this step.

Determine your site's policy for recycled cartridges. Some sites relabel and reuse the cartridges and some sites remove the cartridges from the automated library to use later for accessing historical files.

The recycler executes the `recycler.sh` script when all the current images from a VSN have been re-archived to another VSN.
The recycler calls the `/opt/SUNWsamfs/scripts/recycler.sh` script with the following arguments:

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

For examples of the script, see the `recycler.sh`(1M) man page or view the `/opt/SUNWsamfs/examples/recycler.sh` script. The latter shows how to relabel a recycled VSN and send mail to the superuser.

Configuring Recycling for Disk Archive Volumes

If you are archiving to disk, you must edit the `archiver.cmd` file to recycle.
If you are recycling by archive set, you must add archive set recycling directives between the `params` and `endparams` directives.
If you are recycling by library, this step is optional.

Editing the **archiver.cmd** File

To edit the `archiver.cmd` file, follow the steps described in About the `archiver.cmd` File.
As an alternative, you can edit the `archiver.cmd` file by using the File System Manager. For more information, see the File System Manager online help.

The following table shows the archive set recycling directives that you can use.

Table - Archive Set Recycling Directives

Directive	Function
`-recycle_dataquantity` size	Limits the amount of data the recycler schedules for re-archiving to clear a disk volume of useful data. By default, a limit is ignored for disk archive recycling.
`-recycle_ignore`	Prevents the archive set from being recycled.
`-recycle_mailaddr` mail-address	Sends recycler messages to the specified email address.
`-recycle_mingain` percent	Limits recycling of volumes in the archive set by setting the `mingain` mark for a disk volume. The `mingain` is expressed as a percentage of the expired data associated with the volume. When the expired data of the volume exceeds the `mingain` percentage, the recycler begins to recycle the volume. The default is 50%.
`-recycle_minobs` percent	Limits the recycler's selection of tar files in volume by setting a threshold for the recycler's rearchiving process of disk archive volumes. When the percentage of expired files within an archived tar file on the disk reaches this threshold, the recycler begins moving the current files from the archive into a new tar file. Once all the current files have been moved, the original tar file is marked as a candidate to be removed from the disk archive. The default is 50%.
`-rearch_stage_copy` copy-number	Sets staging for re-archiving to take place from selected (faster) copies.

For more information about archiver directives, see Configuring the Archiver or the `archiver.cmd`(4) man page.

Recycler Logging for Disk Archives

Example - Recycler Log File for Disk Archive Files

```
---Archives---   -----Percent-----
 ----Status-----    Count    Bytes   Use Obsolete Free   Library:Type:VSN
new candidate          0      0        0    41      59  <none>:dk:disk01
677 files recycled from VSN disk01 (mars:/sam4/copy1)
0 directories recycled from VSN disk01 (mars:/sam4/copy1)
```

Recycling for Archive Copy Retention

As an alternative to the normal recycling process, you can use the `sam-nrecycler`(1M) tool to work with the File System Manager's backup and recovery point features. This tool removes expired archive copies and frees archive volumes to aid in the ability to use SAM-QFS dump files for archive retention. To take advantage of this functionality, you must use this recycler in place of the existing `sam-recycler` command.

The `sam-nrecycler`(1M) tool scans file system metadata and SAM-QFS dump files to determine which volumes contain archive images. You can invoke the tool through the `crontab`(1) file at an off-peak time, or at any time using the `sam-nrecycler` command. The nrecycler identifies all archive images on a removable media volume or in a disk archive tar file by scanning all file system `.inodes` files and specified SAM-QFS dump files. The `nrecycler` can then determine whether volumes contain any archive images. The space on these volumes can be reclaimed. If a removable media volume does not contain any archive images, relabel the cartridge. If a disk archive tar file does not contain any archive images, remove the tar file from the disk archive directory.

When `sam-nrecycler`(1M) detects that a removable media volume contains only free or expired space and is safe to relabel, it invokes the `sam-nrecycler.sh` script. The script can relabel the cartridge using either the original VSN or a new VSN. It can then export the cartridge from the library, or it can perform another user-defined action.
When `sam-nrecycler` detects that a disk archive volume contains only free or expired space, it unlinks the unused disk archive tar file.

You control the actions of the `sam-nrecycler`(1M) toll by including directives in the `/etc/opt/SUNWsamfs/nrecycler.cmd` file. You must also specify the path to the directories that contain the SAM-QFS dump files. The list of directories must be complete and all SAM-QFS dump files must be contained in the directory list.

You can also include a `logfile=` directive line in the `nrecycler.cmd` file to specify an nrecycler log file. The system writes recycling messages and recycling reports to this file.

For more information about `sam-nrecycler(1M)`, see the `sam-nrecycler(1M)` man page.

Advanced SAM-QFS Topics

This page discusses advanced topics to Sun Storage Archive Manager (SAM-QFS) system administration and usage.

Using Device Logging

The device-logging facility provides device-specific error information that you can use to analyze certain types of device problems. It can help to determine a failing sequence of events for an automated library, tape drive, or optical drive. The device-logging facility does not collect soft media errors (such as recoverable read errors).

Device-logging messages are written to individual log files. A log file is created for each automated library, for each tape and optical drive device, and for the historian. The log files are located in `/var/opt/SUNWsamfs/devlog`. The name of each log file corresponds to the name of the equipment ordinal.
For example, assume that you have a QFS file system and a single Hewlett-Packard optical library with two optical drives.
The following example shows the `mcf` file.

```
/dev/samst/c1t5u0 40 hp hp40 - /etc/opt/SUNWsamfs/hp40_cat
/dev/samst/c1t4u0 41 mo hp40 -
/dev/samst/c1t6u0 42 mo hp40 -
```

The following example shows the `/var/opt/SUNWsamfs/devlog` file. Device 43 is the historian.

```
# pwd
/var/opt/SUNWsamfs/devlog
# ls
40 41 42 43
#
```

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 great deal of device activity occurs. Initially, the device log settings are set to the following default values:

```
err retry syserr date
```

If you suspect a problem exists with one of the devices, consider enabling additional logging events for that device. Also, 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, your service provider might advise you to set the event to `all` for a device. However, in general, it is not practical to run the system with excessive logging.

The device log information is collected automatically when the `samexplorer`(1M) command is issued. This allows the file system service to review any device error information as part of problem analysis activity.

How to Enable the Device Log By Using the **`samset`**(1M) Command

- Issue the `samset`(1M) command.

  ```
  # samset devlog <eq> <event>
  ```

  For eq, specify the equipment ordinal of the device for which you want to log messages.
  For event, specify one or more of the events listed in the `samset`(1M) man page. If you specify more than one event, separate them with space characters.

How to Enable the Device Log by Editing the **`defaults.conf`** File

1. Become superuser.
2. In the `/etc/opt/SUNWsamfs/defaults.conf` file, add the `devlog` directive.

```
devlog <eq> <event>
```

For eq, specify the equipment ordinal of the device for which you want to log messages.
For event, specify one or more of the events listed in the samset(1M) man page. If you specify more than one event, separate them with space characters.
When a QFS file system starts, it sets the event type for each available device to default. You can also use the samset(1M) command to determine the present settings for each device log.

3. Save and close the defaults.conf file.
4. Use the samd(1M) config command to propagate the defaults.conf file changes.

```
# samd config
```

Using Removable Media Files

You can use the request(1) command to manually create, write, and read files that do not use the disk cache for buffering the data. Files created in this manner are called removable media files.

> Note -
> The request(1) command bypasses the typical functions of the archiver.

Removable media files look like typical QFS files in that they have permissions, a user name, a group name, and size characteristics. However, their data does not reside in the disk cache. Therefore, you can create removable media files that are larger than the disk cache and write them to removable media cartridges.

The system creates an inode entry in the .inodes file for the file that you specify with the request(1) command. The QFS file systems read that information from the inode entry. Multiple removable media files can reside on the same volume.

A removable media file that spans multiple volumes is called a volume overflow file. The volume overflow feature enables a single large file to span multiple volumes on multiple cartridges. The volume overflow feature is useful if you have very large files that exceed the capacity of their chosen media.

You must read and write removable media files sequentially. The QFS file system automatically mounts the requested volume if the volume resides in an automated library defined in the mcf file.

The presence of a removable media file on a volume prevents that volume from being recycled. The recycler expects that only archived files reside on the particular volume that is assigned for archiving. In addition, removable media files are never archived. Removable media files are not supported over NFS.

How to Create a Removable Media or Volume Overflow File

1. Use the tplabel(1M) or odlabel(1M) command to label a tape or magneto-optical cartridge, respectively.
   See Labeling and Unlabeling Cartridges for details.
2. Issue the request(1) command. At a minimum, use the following options:

```
request -m <media-type> -v <vsn> [<vsn>/<vsn> ...] [-l <vsn-file>] <input-file>
```

Argument	Meaning
media-type	The media type of the removable media cartridge. For information about valid media-type specifications, see the mcf (4) man page.
vsn	The volume serial name (VSN) of the removable media cartridge. If you specify more than one VSN, you are creating a volume overflow file. You can specify up to 256 VSNs for volume overflow files. Use forward slash characters (/) to separate the vsn arguments.The VSNs specified should not be among the volumes that are used in a SAM-QFS environment for automated archiving. Archiving appends the next file to be archived to the end of the current data and moves the EOF label beyond the data.

vsn-file	An input file that contains a list of VSNs. When you have many VSNs, use an input file containing the list of VSNs
input-file	The file to be written to the removable media cartridge. This file must reside in a QFS file system.

Examples

The following command creates a removable media file:

```
# request -m lt -v aaa rem1
```

The following command creates a volume overflow file on three volumes:

```
# request -m lt -v TAPE01/TAPE02/TAPE03 large.file
```

For detailed examples of how to create removable media files, see the `request`(1) man page.

Using Segmented Files

The SAM-QFS environment supports segmented files. Segmenting files improves tape storage retrieval speed, access, and manageability for very large files. A segmented file can be larger than the physical disk cache. In this case, only part of a segmented file resides on the disk cache at any one time.

The `segment`(1) command enables you to specify the segment size. You cannot set a segment size that is smaller than the current file size.

Segmented files support tape striping. After a file is segmented, it can be striped simultaneously over multiple tape devices, which significantly reduces the time needed to store the file segments. Data access is accelerated by allowing users to retrieve only the desired file segments rather than the entire file.

Segmentation can enhance archiving efficiency because only changed portions of a file are re-archived. Segments of a file can be archived in parallel, and segmented files can be staged in parallel. This increases performance during archiving and retrieving.

Segmentation can be enabled on a file, directory, or entire file system. Segmented files support all other SAM-QFS capabilities.

> Note -
> The `mmap` function cannot take place on a segmented file. Therefore, a segmented file cannot be an executable binary.

The following sections describe how segmented files differ from nonsegmented files. For more information about segmented files, see the `segment`(1) or the `sam_segment`(3) man pages.

Archiving

For a segmented file, the archivable unit is the segment itself, not the file. All archiving properties and priorities apply to the individual segment and not to the file.

You can stripe a segment by specifying both the `- drives` and `-drivemin` parameters for the archive set in the `archiver.cmd` file. For example, assume that a 100-megabyte segmented file in the file system has segment size of 10 megabytes. If the `archiver.cmd` file defines an archive set with a `-drives` 2 directive, this file is archived to two drives in parallel. Segments 1, 3, 5, 7, and 9 are archived using the first drive, and segments 2, 4, 6, 8, and 10 are archived using the second drive.

Only segments that have been modified are archived. Up to four archive copies can be made for each segment. SAM-QFS also supports volume overflow for segments.

> Note -
> The index of a segmented file contains no user data. It is considered metadata and is assigned to the file system archive set.

Disaster Recovery

For information about recovering a segmented file in the event of a disaster, see SAM-QFS Troubleshooting.

Using System Error Facility Reporting

The system error facility (SEF) reporting system captures log sense data from tape devices in an automated library, writes it to a log file, and translates it into human-readable form. This utility consists of the following:

- A log file containing data from tape device log sense pages.
- A command, `sefreport`(1M), for writing the log file to `stdout` in a human-readable format. This log file can be used as input to a user-supplied analysis script.
  The log sense pages are different from vendor to vendor. For the meanings of the parameter codes, control bits, and parameter values, see the vendor documentation for each specific device.

SEF is not supported for stand-alone tape drives. SEF reporting is most useful for older SCSI-2 devices that do not support the `tapealert`(1M) functionality. For more information, see the `tapealert`(1M) man page.

How to Enable SEF Reporting

1. Become superuser.
2. Issue the `mkdir`(1) command to create the SEF directory. For example:

```
# mkdir /var/opt/SUNWsamfs/sef
```

3. Use the `touch`(1) command to create the log file.
   You can enable SEF reporting any time after installation by creating the `sefdata` log file. Initially, the SEF log file must be empty. The following command shows the SEF log file being created in the default location.

```
# touch /var/opt/SUNWsamfs/sef/sefdata
```

4. Use the `samd stop` and `samd start` commands to initialize SEF reporting.

```
samcmd aridle
samcmd stidle
```

Wait until the tape drives are idle. Then unload the tape drives:

For eq, specify the Equipment Number of the drive.

1. Use the `samd`(1M) `unload` command to unload all removable media:

```
samcmd unload <eq>
```

Repeat the above commands for each tape drive.

```
# samd stop
# samc config
# samd start
```

SEF data is appended to the log file as it is generated.

> Note -
> SEF reporting is enabled as long as the `sefdata` log file exists. To disable SEF reporting, rename or remove this file.

You can configure SEF reporting to log and read log sense data from an alternate location. For more information about reading log sense data from an alternate location, see the `sefreport`(1M) man page.

How to Generate SEF Report Output

The SEF report output consists of header lines and log sense data.

1. Verify that `/opt/SUNWsamfs/sbin` is in your command path.
2. Use the `sefreport`(1M) command to generate SEF output.
   The following are the most commonly used options with the `sefreport`(1M) command:

- The `-d` option. The `-d` option generates additional device information. It writes an additional header line that contains the equipment ordinal and path name to the device for each record. This process makes searching for and locating SEF records that pertain to a specific device easier.
- The `-v` option or the `-t` option. Do not specify the `-t` and `-v` options on the same command line. They are mutually exclusive.
  - The `-v` option generates information in verbose mode. It appends information regarding the equipment ordinal, page code, and VSN to each line of a record. This method enables selecting only those lines that pertain to a specific device or a specific volume.
  - The `-t` option generates log sense output with text descriptions. For each line of log sense data output, the report includes an additional string containing the equipment ordinal, page code, VSN, and parameter code description.
    For example, the following SEF command reads the SEF log file from the default location, writes the device number and path name for each device, and generates output:

```
# sefreport -d /var/opt/SUNWsamfs/sef/sefdata > sef.output
```

Example: Content of sef.output File

```
Record no. 1
Mon Mar 26 11:17:48 2001 STK 9840 1.25 VSN 002981
Eq no. 32 Dev name /dev/rmt/1cbn
PAGE CODE 2
param code control param value
00h 74h 0x0
01h 74h 0x0
02h 74h 0x0
03h 74h 0x0
04h 74h 0x0
05h 74h 0x40050
06h 74h 0x0
PAGE CODE 3
param code control param value
00h 74h 0x0
01h 74h 0x0
02h 74h 0x0
03h 74h 0x0
04h 74h 0x0
05h 74h 0x140
06h 74h 0x0
PAGE CODE 6
param code control param value
00h 74h 0x0
Record no. 2
Mon Mar 26 11:30:06 2001 STK 9840 1.25 VSN 002999
Eq no. 31 Dev name /dev/rmt/0cbn
PAGE CODE 2
param code control param value
00h 74h 0x0
01h 74h 0x0
02h 74h 0x0
03h 74h 0x0
04h 74h 0x0
05h 74h 0x1400a0
06h 74h 0x0

PAGE CODE 3
param code control param value
00h 74h 0x0
01h 74h 0x0
02h 74h 0x0
```

```
03h 74h 0x0
04h 74h 0x0
05h 74h 0x190
06h 74h 0x0
PAGE CODE 6
param code control param value
00h 74h 0x0
Record no. 3
Mon Mar 26 11:30:23 2001 STK 9840 1.25 VSN 002981
Eq no. 32 Dev name /dev/rmt/1cbn
PAGE CODE 2
param code control param value
00h 74h 0x0
01h 74h 0x0
02h 74h 0x0
03h 74h 0x0
04h 74h 0x0
05h 74h 0x18400f0
06h 74h 0x0

PAGE CODE 3
param code control param value
00h 74h 0x0
01h 74h 0x0
02h 74h 0x0
03h 74h 0x0
04h 74h 0x0
05h 74h 0x1e0
06h 74h 0x0
PAGE CODE 6
param code control param value
00h 74h 0x0
.
```

```
.
.
```

For more information about the SEF log file, including its content and format, see the sefdata(4) man page. For more information about optional SEF report formats, see the sefreport(1M) man page.

Managing the SEF Log File

You manage the SEF log file just as you manage any other SAM-QFS log file. You can run a cron(1) job periodically to save the current log file to another location, to delete old SEF files, to create new (empty) SEF files, or to perform other file management tasks.
You can also use the log_rotate.sh(1M) utility to rotate this log file.
For more information about tools for managing the SEF log file, see the cron(1) or log_rotate.sh(1M) man pages.

SEF **sysevent** Functionality

In addition to using the SEF log file, you can use the Solaris sysevent feature to obtain tape drive SCSI log sense error counter pages 2 and 3 for media analysis. By default, the SEF sysevent feature is enabled and set to poll once before unload. The SEF sysevent behavior is controlled by defaults.conf and samset.

In the defaults.conf file, the sef parameter can be used to enable SEF sysevent feature by equipment ordinal, or to specify the log sense polling frequency. For more information, see the defaults.conf(4) manu page.

How to Create the SEF **sysevent** Handler

1. Create a /var/tmp/xx file similar to the following.

```
#!/bin/ksh
echo "$@" >> /var/tmp/xx.dat
exit 0
```

2. Make the /var/tmp/xx file executable.

```
# chmod a+rwx /var/tmp/xx
```

   Add the SEF sysevent handler to the syseventd file by adding the following information.

```
# syseventadm add -vSUNW -pSUNWsamfs -cDevice -sSEF
/var/tmp/xx \"\$VENDOR\" \"\$PRODUCT\" \"\$USN\" \"\$REV\" \
$TOD \$EQ_ORD \"\$NAME\" \$INQ_TYPE \"\$MEDIA_TYPE\" \"\$VSN\" \
\$LABEL_TIME \$LP2_PC0 \$LP2_PC1 \$LP2_PC2 \$LP2_PC3 \$LP2_PC4 \
\$LP2_PC5 \$LP2_PC6 \$LP3_PC0 \$LP3_PC1 \$LP3_PC2 \$LP3_PC3 \
\$LP3_PC4 \$LP3_PC5 \$LP3_PC6 \$WHERE \$sequence
# syseventadm restart
```

   This command creates the /etc/sysevent/config/SUNW,SUNWsamfs,Device,sysevent.conf file containing the SEF sysevent handler /var/tmp/xx and loads the event handler into the syseventd daemon.

3. To load the SEF sysevent handler, issue the following command:

```
pkill -HUP syseventd
```

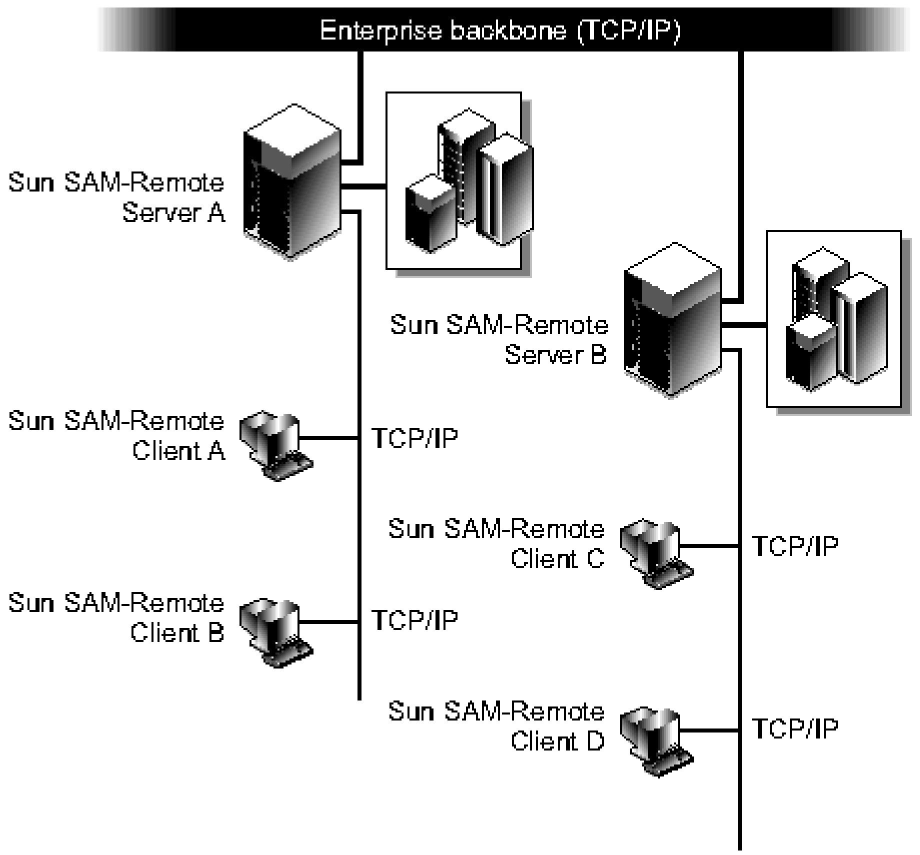
   For more information, see the sefsysevent(4) man page.

Using the Sun SAM-Remote Software

The Sun SAM-Remote client and the Sun SAM-Remote server form an implementation that allows libraries and other removable media devices to be shared between SAM-QFS host systems. Use the SAM-Remote software to configure multiple storage clients that archive and stage files from a centralized tape library or magneto-optical library. For example, if you have host systems on a network that spans a large geographical area, files created in one city can be archived to cartridges in a library located miles away.

Sun SAM-Remote Software Overview

Figure - Two Sun SAM-Remote Servers, Each With Two Clients



Sun SAM-Remote software provides the following advantages:

- Enables you to configure remote sharing of an expensive removable media resource, such as a library, between two or more Sun SAM-Remote clients.
- Enables clients to migrate data to a server.
- Enables multiple SAM-QFS servers to be hosts to one another. In a Sun SAM-Remote environment, the server is the host system that is configured with an equipment type of `ss` in the `mcf` file.

You can configure the Sun SAM-Remote server and clients to provide multiple archive copies between two or more Sun Solaris host systems. For example, you can configure two Solaris systems running SAM-QFS software as both Sun SAM-Remote servers and Sun SAM-Remote clients to each other. Benefits of this configuration include the ability to create local copies for each server with an additional archive copy of data on the other server. File systems can be shared between servers using standard NFS. In the event of a loss of access to the local library, Sun SAM-Remote software would automatically retrieve file data from the archive copy. Users of both servers would have uninterrupted access to their data even if their primary storage library were unavailable.

System Requirements

Before attempting to configure a Sun SAM-Remote environment, make sure that your environment includes the following software and hardware:

- SPARC® or x64 systems with licensed, installed, and operable SAM-QFS archive management software packages.

- Host systems with identical SAM-QFS software revision levels and identical patch collections installed. If some host systems have to be upgraded, see Upgrading QFS and SAM-QFS.
- One or more host systems to act as the Sun SAM-Remote server with at least one SAM-QFS file system.
- A network connection running a TCP/IP connection between the clients and the server.

Software Limitations

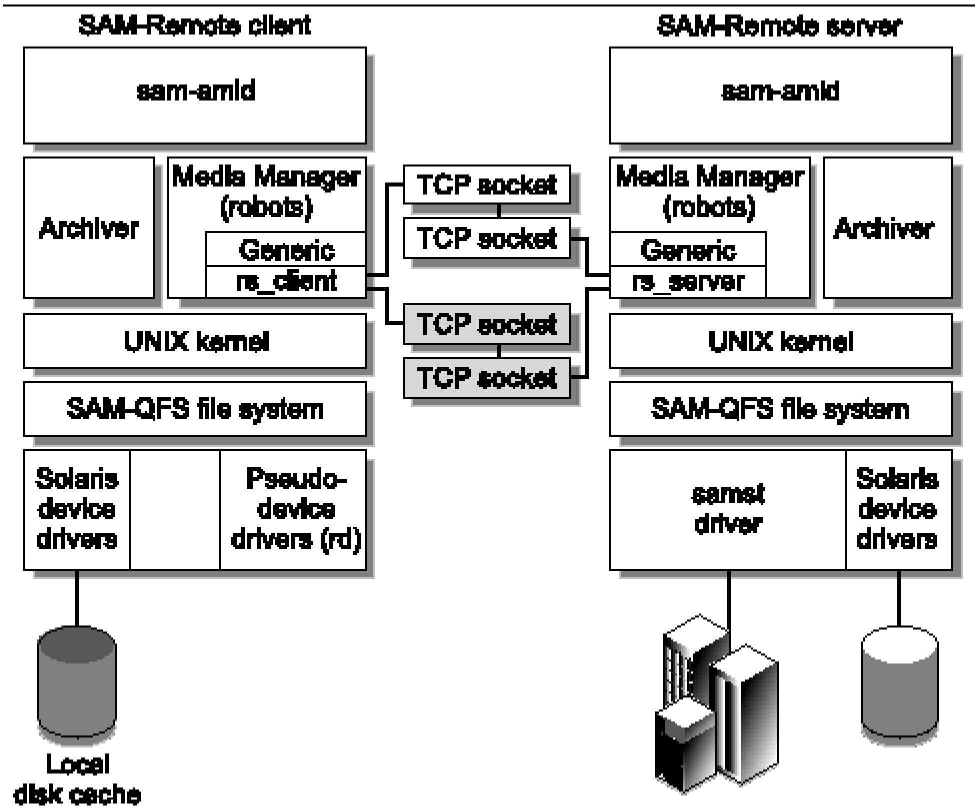
The SAM-QFS software treats cartridges in a remote library no differently than it treats cartridges in a local library. The following information, however, indicates the limits of Sun SAM-Remote software:

- You can recycle media using Sun SAM-Remote, but you attempt this activity only after thoroughly testing your environment. For more information, see Recycling With the Sun SAM-Remote Software.
- Only one daemon on a Sun SAM-Remote client can communicate to the Sun SAM-Remote server.
- SAM-QFS software, and therefore Sun SAM-Remote, cannot operate on Sun QFS clients in a shared Sun QFS file system. When running on a server that is a metadata server for some Sun QFS file systems and a client for other Sun QFS file systems, SAM-QFS software and Sun SAM-Remote operate only on the file systems for which that server is a metadata server.

Client and Server Interactions

Sun SAM-Remote clients interact with the Sun SAM-Remote server using a TCP/IP connection. The network between Sun SAM-Remote clients can be any network type supported by the Solaris OS, such as Ethernet, Fast Ethernet, or Fibre Channel.

Figure - Sun SAM-Remote Server and Client Interactions



Sun SAM-Remote Server Overview

The Sun SAM-Remote server consists of a full-capability SAM-QFS storage management host and a Sun SAM-Remote server daemon that defines libraries to be shared among the clients. At least one SAM-QFS file system must be configured on the Sun SAM-Remote server.

You define a host system as a Sun SAM-Remote server by adding a line in the server system's `/etc/opt/SUNWsamfs/mcf` file with an equipment type of `ss`. You must provide a unique family set name for each server. Up to ten clients can be configured per daemon. To configure more than ten clients, add an additional remote server entry in the `mcf` file for each ten clients that you want to configure. For more information about the server daemon, see the `sam-remote`(7) man page.

Sun SAM-Remote Client Overview

The Sun SAM-Remote client is a SAM-QFS host system that establishes a Sun SAM-Remote client daemon containing a number of pseudo-devices.

You define a host system as a Sun SAM-Remote client by adding a line in the client system's `/etc/opt/SUNWsamfs/mcf` file with an equipment type of `sc`. For more information about the client daemon, see the `sam-remote`(7) man page.

A pseudo-device defines a network connection to an actual removable media device on the Sun SAM-Remote server. Pseudo-devices have an equipment type of `rd`, which is an abbreviation for remote device. You define the pseudo-devices in the Sun SAM-Remote client's `/etc/opt/SUNWsamfs/mcf` file. The Sun SAM-Remote daemon and pseudo-devices are associated with one particular server.

The Sun SAM-Remote daemon supports an unlimited number of pseudo-devices for each client. The actual number of pseudo-devices to be used by the client is configurable. When determining how many pseudo-devices should be configured per client, think of these devices as the number of simultaneous data transfers that can occur between the client and the server. As more pseudo-devices are defined, the possibility of increasing the total network traffic load increases. As the system administrator, determine the actual number of pseudo-devices needed for the system.

Interaction Between the Sun SAM-Remote Server and the Sun SAM-Remote Client

The Sun SAM-Remote server daemon, `sam-serverd`, listens for the clients on port 1000. You can configure a different port in the Sun Solaris `/etc/services` directory with a service name of `rmtsam`. When a Sun SAM-Remote client connects to the Sun SAM-Remote server, the `sam-serverd` daemon establishes a connection on another port and communicates this port number to that client, using the defined port. The socket size is passed to the client. The socket size is configurable and is described in more detail in Configuring the Sun SAM-Remote Software.

Library Catalogs

The Sun SAM-Remote library catalog is a subset of the catalog located on the Sun SAM-Remote server. The client catalog is updated in real time. The slots allotted to a Sun SAM-Remote client catalog are controlled only by the Sun SAM-Remote server.

After initialization, the system builds a client catalog and passes it to the Sun SAM-Remote client based on information from the Sun SAM-Remote server catalog file. After the connection between the host and client is established, media available to the client is flagged as available. If the connection between the client and server is lost, the media on the client side is flagged as unavailable. You can view the media availability through the `samu`(1M) v display. The information that appears in the `samu`(1M) v display on the client is a subset of what appears in the v display on the server. A good practice is to access the media catalog through the `samu`(1M) v display on Sun SAM-Remote server. For more information about the Sun SAM-Remote server client file, see Configuring the Sun SAM-Remote Software. For information about the `samu`(1M) operator utility, see Using the samu Operator Utility.

Changes to the catalog are passed between hosts as necessary. Any changes in the server catalog that involve a media type associated with a client are passed to the client, and the client catalog is updated.

Archiving

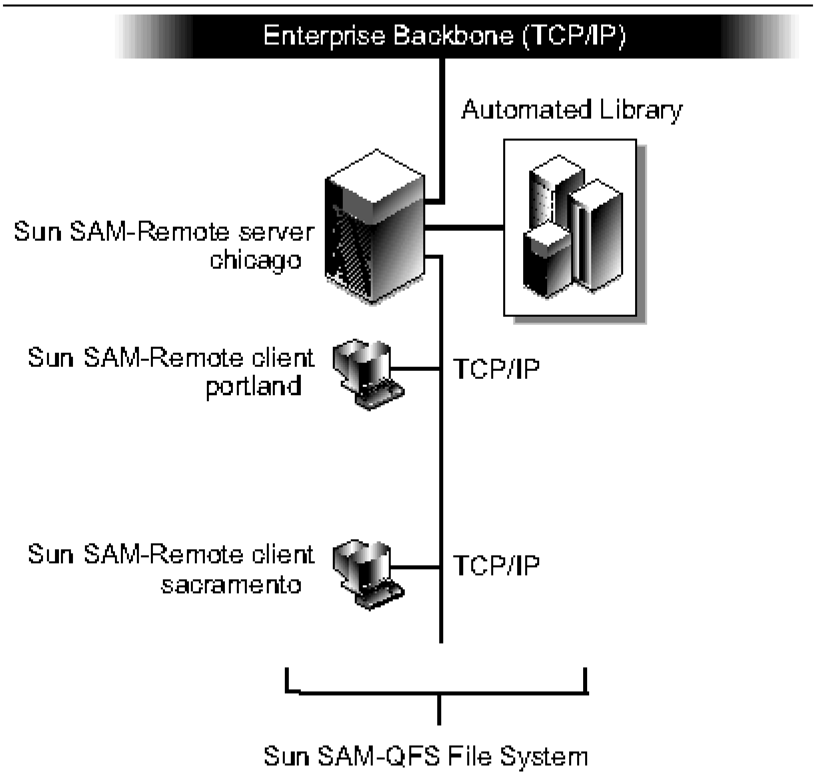
Sun SAM-Remote archive processing is the same as SAM-QFS archive processing. The Sun SAM-Remote client makes a mount request to be added to the server's mount request table. The client then waits for the server to respond with a message indicating that the media is mounted. Archiving begins when the media is available.

Configuring the Sun SAM-Remote Software

This section explains how to perform an initial configuration of the Sun SAM-Remote server and client software.

Figure - Example Sun SAM-Remote Configuration

The SAM-QFS file systems on `portland` and `sacramento` are clients of the Sun SAM-Remote server on `chicago`.



The following procedures explain how to configure the Sun SAM-Remote software on a Sun SAM-Remote server and on one or more Sun SAM-Remote clients, using the example configuration.

- Perform the procedures in the order described.
- You must have superuser access to the server system on which the Sun Sun SAM-Remote software is to be installed.
- You must have superuser access to the client system or systems on which the Sun SAM-Remote software is to be installed.
- The same release and revision level of SAM-QFS software must be installed on all client and server hosts in the Sun SAM-Remote environment.
- The same patch collection must be installed on all client and server hosts in the SAM-Remote environment.

Step 1: Log In to the Potential Server and Client Hosts

1. Log in to the Sun SAM-Remote server as the superuser.
2. Log in to the Sun SAM-Remote clients as the superuser.

Step 2: Verify Client and Server Software

Follow these steps on each system to be configured as part of a Sun SAM-Remote environment.

1. Issue the `pkginfo`(1M) command with its `-l` option, and examine the output.

```
chicago#  pkginfo -l SUNWsamfsr
   PKGINST:  SUNWsamfsr
      NAME:  Sun SAM and Sun SAM-QFS software Solaris 10 (root)
  CATEGORY:  system
      ARCH:  sparc
   VERSION:  5.0.7,REV=5.10.2009.04.09
   BASEDIR:  /
    VENDOR:  Sun Microsystems, Inc.
      DESC:  Storage and Archive Manager File System
    PSTAMP:  wiffleball-mn20090409140022
  INSTDATE:  Jul 29 2009 11:00
   HOTLINE:  Please contact your local service provider
    STATUS:  completely installed
     FILES:      634 installed pathnames
                  23 shared pathnames
                  24 linked files
                  78 directories
                 199 executables
                   1 setuid/setgid executables
               74889 blocks used (approx)


chicago# pkginfo -l SUNWsamfsu
   PKGINST:  SUNWsamfsu
      NAME:  Sun SAM and Sun SAM-QFS software Solaris 10 (usr)
  CATEGORY:  system
      ARCH:  sparc
   VERSION:  5.0.7,REV=5.10.2009.04.09
   BASEDIR:  /
    VENDOR:  Sun Microsystems, Inc.
      DESC:  Storage and Archive Manager File System
    PSTAMP:  wiffleball-mn20090409140036
  INSTDATE:  Jul 29 2009 11:01
   HOTLINE:  Please contact your local service provider
    STATUS:  completely installed
     FILES:       57 installed pathnames
                  17 shared pathnames
                  20 directories
                  19 executables
                9597 blocks used (approx)


chicago#
```

The output shows that the server (`chicago`) is running software version 4U0.5. Any systems included in an environment with this server must also run version 4U0.5.

2. Issue the `showrev`(1M) command with its `-p` option, and examine the output.

```
chicago# showrev -p | grep SUNWsamfs

chicago#
```

The output shows that the server has no SAM-QFS patches installed. Any systems included in an environment with this server must also run version and patch level.
If you need to perform any software upgrades, see Upgrading QFS and SAM-QFS.

Step 3: Edit the `mcf` Files on Each Client

The `mcf` file defines a file system. This procedure adds the definition of the host as a Sun SAM-Remote client.

1. From the Sun SAM-Remote server, stop the SAM-QFS functions.
2. Issue the `samcmd`(1M) command to idle each removable media. For more information about the `samcmd`(1M) command, see the `samcmd`(1M) man page.

```
# samcmd idle <eq1>
# samcmd idle <eq2>
.
.
# samcmd idle <eqn>
```

where eq specifies the equipment ordinal of the removable media drive, as defined in the `mcf` file.
As an alternative, you can also use the `samu`(1M) operator utility to idle the drives. For information, see Using the samu Operator Utility.

> Note -
> All drives must be idle before you issue the next command, `samd stop`, so that the archiver, stager, and other processes can complete current tasks, and cartridges can be unloaded and put into their storage slots.

3. Issue the `samd`(1M) command with its `stop` option to stop the `sam-amld` daemon and its child processes.

```
# /opt/SUNWsamfs/sbin/samd stop
```

4. On each client, open the `/etc/opt/SUNWsamfs/mcf` file.
5. Define each system as a client of the Sun SAN-Remote server, following the example. The example code shows the client `portland`'s `mcf` file after it is edited to make `portland` a client of the Sun SAM-Remote server `chicago`.
6. Open the `mcf` file of another client. In the example, the client `sacramento`'s `mcf` file is edited.
7. Copy the last set of lines from the first client's `mcf` file to next client's `mcf` file. In the example, copy the last set of lines from `portland`'s `mcf` file to `sacramento`'s `mcf` file.
8. Save and close the `mcf` files.

Example: *mcf* Files on the Clients

```
# mcf file on portland
#
# Define a Sun QFS file system
#
# Equipment                                     Eq  Eq Family Dev Additional
# Identifier                                    Ord Ty Set    St  Parameters
# ==========                                    === == ====== == ==========
samfs1                                          10  ms samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE5F00048F2Dd0s6 11  md samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE23000B24C2d0s6 12  md samfs1 on


# Define Sun SAM-Remote Client portland to Sun SAM-Remote server chicago
#
/etc/opt/SUNWsamfs/rmt200 200 sc chicagoss on /var/opt/SUNWsamfs/catalog/tcat
/dev/samrd/rd0            201 rd chicagoss on
/dev/samrd/rd1            202 rd chicagoss on
```

The `mcf` entry on the client consists of a single-line entry for the Sun SAM-Remote client and a pseudo-device entry, indicated by the `rd` equipment type, for each device you want to configure. A pseudo-device defines a network connection to an actual device on the Sun SAM-Remote server. Each entry uses the following fields:

Equipment Identifier	The full path name of the client configuration file. You create the client configuration file in Step 4.
Eq Ord	The number that identifies this client system. Use this number to specify the eq parameter.
Eq Ty	A two-letter mnemonic that identifies the type of the client system.

Family set	The family set name of the daemon to be used on this server. A Sun SAM-Remote server can have one server daemon per client. Note - The family set name on the sam-remote client must match the family set name on the sam-remote server.
Dev St	Device state: on or `off`
Additional parameters	An optional field. In this example, it is the path to the catalog file.

Step 4: Create a Sun SAM-Remote Client Configuration File

A Sun SAM-Remote client's configuration file contains one entry: the name of the Sun SAM-Remote server.

1. On each client, create the file to be used as the Sun SAM-Remote client configuration file in the location that you specified in the `mcf` file. For example:

```
portland# vi /etc/opt/SUNWsamfs/rmt200
```

2. Type the name of the Sun SAM-Remote server.
The following example shows the client configuration file for the client `portland`, specifying that its Sun SAM-Remote server is the `chicago` system.

Example: Client Configuration File

```
portland# cat /etc/opt/SUNWsamfs/rmt200
chicago
```

Step 5: Edit the Server's `mcf` File

1. On the Sun SAM-Remote server, edit the `/etc/opt/SUNWsamfs/mcf` file.
2. Add the entries that define the SAM-QFS file systems and this system as the Sun SAM-Remote server. You must have at least one SAM-QFS file system. The following example shows the `mcf` file on `chicago`.

Example: *`mcf`* Files on the Server

```
# mcf file on Sun SAM-Remote server chicago:
# Define a SAM-QFS file system
#

# Equipment                                         Eq  Eq Family Dev Additional
# Identifier                                        Ord Ty Set    St  Parameters
# ==========                                        === == ====== == ==========
samfs1                                              10 ms samfs1 on
/dev/dsk/c6t600A0B80004850A600000F8048EF90ADd0s6 11 md samfs1 on
/dev/dsk/c6t600A0B800048505600000E9D48EF91EEd0s6 12 md samfs1 on


# Define a tape library that client portland can use:
/dev/samst/c4t500104F0009C2F6Fu0 100 sn rb100 on /var/opt/SUNWsamfs/catalog/rb100.cat
/dev/rmt/0cbn                    101 li rb100 on
/dev/rmt/1cbn                    102 li rb100 on


# Define Sun SAM-Remote server chicago
#
/etc/opt/SUNWsamfs/rmt200         50 ss chicagoss on
```

Note -
The family set name on the sam-remote server must match the family set name on the sam-remote client.

Step 6: Create the Server's Configuration File

The Sun SAM-Remote server configuration file defines the disk buffer characteristics and media to be used for each client. Ten clients can be configured per server daemon. To support more clients, configure another Sun SAM-Remote server daemon.

1. On the server, create the Sun SAM-Remote server configuration file.
2. Add definitions of each client, using the following format:

```
<client-name>
[ <parameter1> ]
    media
    <eq> <media-type> <regex>
     [<eq> <media-type> <regex> ]
     [. . .]
   endmedia
```

Field	Definition and Requirements
client-name	Identifies the system to be served by this invocation of the Sun SAM-Remote daemon. Use the network name, its IP address, or a fully-qualified domain name. The first character in client-name must be the first character in the line.
parameter (optional)	Defines an attribute of the client, using a keyword = value pair. For example, you can use the parameter to specify the network block size to be used by this client's socket in kilobytes. The format for this parameter: `net_blk_size=<size>` where size is an integer from 4≤ size≤ 64. The default is 4, which specifies 4096 bytes. The line containing the parameter must start with space or tab characters.
`media` and `endmedia`	Keywords that contain the media definitions. The definitions within these two keywords define the media archive volumes that a client can use. Note - These keywords are required and must be indented with space or tab characters.

eq media-type (regex)	Defines a media archive volume that this client can use. Enclose the regex data with parentheses. Because network-attached libraries have mixed media, specify each media type on a separate line. Note - Use the space or tab characters to indent the media definitions. * eq: Equipment Number as shown in the `mcf` file. * media-type: Two-character specific media type, such as `li`. For information about valid media types, see the `mcf`(4) man page but do not use the generic media type. * (regex): The volume serial names (VSN) of the cartridges to which the files are archived, expressed as an extended regular expression. For information about extended regular expressions, see the `egrep`(1) man page. For information about regular expressions, see the `regcomp`(3C) man page.

For example, the following is a valid media type definition:

```
   media
      100 li (VSN1)
      100 li (VSN2)
   endmedia
```

Note -
Do not allow the same physical media cartridges to be used by more than one client. Also, if the Sun SAM-Remote server has its own file system outside of the Sun SAM-Remote environment, it is not recommended that a cartridge be used by both the client and the server.

The following example shows the server configuration file, `/etc/opt/SUNWsamfs/rmt200` for the Sun SAM-Remote server `chicago`. This file defines clients `portland` and `sacramento`.

Example: Server Configuration File

```
# Sun SAM-Remote server config file /etc/opt/SUNWsamfs/rmt200
#
portland
   media
      100 li (100031|100032|100034|100035|100037|100038)
      100 li (200001|200002|200003|200004|200005|200006)
   endmedia
#
#
sacramento
   media
      100 li (300141|300142|300143|300145|300146|300147)
      100 li (400001|400002|400003|400005|400006|400007)
   endmedia
```

Step 7: Enable Archiving

1. Verify the `archiver.cmd` file on each client. Depending on your configuration, you might need to perform the following tasks:
   a. Make sure that the VSNs that are defined in the server configuration file are assigned to the correct archive sets.
   b. Remove the following directives if they apply to archive sets to be archived in the library connected to the Sun SAM-Remote server:
      - `-tapenonstop`
      - `-offline_copy direct`
2. Issue the samd(1M) command with its `start` option to start the SAM-QFS processes on the server and on the clients. Enter the following command on the clients and the server:

```
server# samd start
```

3. Check the status of the Sun SAM-Remote connections.
   a. Issue the samu(1M) command on the server and the clients. For more information, see the samu(1M) man page or Using the samu Operator Utility.
   b. On each client, view the the samu(1M) utility's (s) display.
   The following example shows the status s display on the Sun SAM-Remote client portland. The device type sc identifies the Sun SAM-Remote client. The message after that line indicates that a connection with the server chicago has been established.

Example: ***samu }}({{s***) Display for a Client

```
Device status samcmd      5.0.7 20:44:09 Jul 30 2009
samcmd on portland

ty     eq state   device_name                        fs status
sc     200 on     /etc/opt/SUNWsamfs/rmt200          200  ---------r
        Remote server 10.1.229.92 connected
rd     201 on     /dev/samrd/rd0                     200  ----------

rd     202 on     /dev/samrd/rd1                     200  ----------

hy     203 on     historian                          203  ----------
```

   c. On the server, view the the samu(1M) utility's (s) display.
   The following example shows the samu(1M) status s display on the Sun SAM-Remote server chicago. The device type ss identifies the Sun SAM-Remote server.

Example: ***samu }}({{s***) Display for a Server

```
Device status samcmd      5.0.7 20:40:05 Jul 30 2009
samcmd on chicago

ty      eq state   device_name                        fs status
sk       1 on      /etc/opt/SUNWsamfs/SL500_SAM         1  m--------r
        running
li       2 on      /dev/rmt/0cbn                        1  ---------p
        empty
li       3 on      /dev/rmt/1cbn                        1  ---------p
        empty
ss      50 on      /etc/opt/SUNWsamfs/rmt200           50  -------o-r

hy      51 on      historian                           51  ----------
```

   d. On the server, view the the samu(1M) utility's (R) display.
   The following example shows the samu(1M) Sun SAM-Remote R display from the Sun SAM-Remote server chicago.

Example: ***samu*** (***R***) Display for a Server

```
Remote server eq: 50    addr: 00003858  samu      5.0.7 20:41:38 Jul 30 2009


message:

Client IPv4: jimmy 192.10.10.3                                  port - 5000
          client index - 0  port - 0 flags - 0004

Client IPv4: portland 10.1.229.97                               port - 5000
          client index - 1  port - 32848 flags - 0005  connected
```

If the Sun SAM-Remote configuration includes several clients, press the CONTROL-f key sequence to scroll through each client. Each client is identified by name and by its client index field, an integer 0 - 9, which indicates its order in the possible 10 clients defined for this server daemon. The network block size, maximum file size, and minimum file size are listed in bytes. Flags indicate the state of the connection:

Flag	Number	Meaning
0x00000000	0004	No connection.
0xc0000000	0005	A connection has been established.

4. On the server, use the samu(1M) utility's v display to ensure that the Sun SAM-Remote catalog is available on the clients. From samu (1M), enter the following:

```
:v <eq>
```

For eq, specify the equipment ordinal of the Sun SAM-Remote client daemon as defined in the mcf file.

Example: ***samu*** (***v***) Display

The example shows the volumes that portland can access.

```
Robot VSN catalog by slot : eq 200 samu 4.0.5 Wed May 02 15:24:13
count 32
slot access time count use flags ty vsn
1 2003/01/02 10:40 0 0% -il-o-b-R-U- at 000032
2 2003/01/02 11:41 0 0% -il-o-b-R--- at 000034
3 2003/01/02 12:42 170 91% -il-o-b----- at 000035
4 2003/01/02 13:43 20 7% -il-o-b----- at 000037
5 2003/01/02 14:44 0 0% -il-o-b----- at 000038
6 2003/01/02 13:41 0 0% -il-o-b----- at 000031
```

5. From each client, issue the archiver(1M) command and its -A option to verify that archiving is taking place from the client to the server. This command writes a listing from the archiver, including the VSNs from the server. For information about this command, see the archiver(1M) man page.
If files are not archiving, see SAM-QFS Troubleshooting.

Recycling With the Sun SAM-Remote Software

This section contains information about recycling with Sun SAM-Remote. Sun Microsystems recommends recycling in a Sun SAM-Remote environment only under the very specific circumstances described here.

Because the recycling process involves freeing space on cartridges for more data, it is possible for the recycler to destroy important data on archive cartridges if the recycling process is not configured properly.

Note -
These restrictions are not enforced by the SAM-QFS software.

To avoid data loss, it is essential that you adhere to the following restrictions:

- Before using the recycler in a Sun SAM-Remote environment you must have a complete understanding of each step of the recycler. Executing commands in the wrong order, or on the wrong system, can result in an irreversible loss of data. Make sure you have analyzed a command's actions before executing any command, such as tplabel(1M), that can delete data on the Sun SAM-Remote client or the Sun SAM-Remote server.
- Recycling activities on the Sun SAM-Remote server and the Sun SAM-Remote client must not overlap. The result could be accidental relabeling of cartridges and irreversible loss of data.
- You must not recycle cartridges that contain removable media files.
- In a Sun SAM-Remote client and server environment, the client and server are unaware of each other's file systems, data files, and inode files. Therefore, the server and the client each must have exclusive use of a certain set of cartridges. Neither must ever use the other's cartridges.

You can prevent accidental recycling of VSNs used by Sun SAM-Remote clients by creating a `no_recycle` list in the Sun SAM-Remote server's `/etc/opt/SUNWsamfs/ recycler.cmd` file. However, be careful of using the `chmed`(1M) command's `+c` option on volumes in a `no_recycle` list. When you use this command to set the recycling flag (`+c`) on a volume, that action overrides the `no_recycle` list in the `/etc/opt/SUNWsamfs/recycler.cmd` file.
- You must not attempt to recycle volumes on the Sun SAM-Remote server and Sun SAM-Remote client on the same day.

Recycling in a Sun SAM-Remote environment is allowed to occur only if the following conditions are present:

- Each VSN in the system is used by one client system or by the server. There cannot be files from multiple systems on any VSN.
- No Sun SAM-Remote client has catalog entries for any VSNs other than those VSNs containing that client's archive images. The regex values in the server configuration file's media definition lines (the eq media-type regex lines) must agree with the volumes specified in the client catalog. In addition, the regex specifications in the client catalogs cannot specify the same volumes.
- The archiving is performed on an archive set basis. When you are using Sun SAM-Remote, recycling must be performed by archive set, not by library.

The following subsections describe two methods for enabling recycling using a Sun SAM-Remote client and server. The methods are as follows:

- Recycling in a Sun SAM-Remote Environment--Method 1
- Recycling in a Sun SAM-Remote Environment--Method 2

Recycling in a Sun SAM-Remote Environment---Method 1

The procedures in this section describe one method for enabling recycling in a Sun SAM-Remote environment in which the server is named `sky` and the client is named `zeke`.

> Caution -
> To use the recycler in a Sun SAM-Remote environment, you must follow this procedure completely and you must test your configuration to verify that recycling is taking place correctly.

Configuration Files for the Server

The server must have Sun SAM-Remote configuration information in its `mcf` file and in its server configuration file. The following examples show these files.

Example: The ***mcf*** File on Server ***jimmy***

```
# This is the mcf file for the server (jimmy).
# The server parameters file (rmt2000) points
# back to the correct automated library's equipment number
# (1000) for the SL8500 tape library.
#

# Equipment                                       Eq  Eq Family Dev Additional
# Identifier                                      Ord Ty Set    St  Parameters
# ==========                                      === == ====== == ==========
samfs1                                             100 ma samfs1 on
/dev/dsk/c6t600A0B80004850A600000F8048EF90ADd0s0  101 mm samfs1 on
/dev/dsk/c6t600A0B800048505600000E9D48EF91EEd0s6  102 mr samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE5F00048F2Dd0s6  103 mr samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE23000B24C2d0s6  104 mr samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE50000ADAECd0s6  104 mr samfs1 on

samfs2                                             200 ms samfs2 on
/dev/dsk/c7t60003BA13F71500048EDCE720001B17Fd0s6  201 md samfs2 on
/dev/dsk/c7t60003BA13F71500048EDCE7200014BEAd0s6  202 md samfs2 on

# SL8500
/etc/opt/SUNWsamfs/T10K        1000    sk      T10K    on      /etc/opt/SUNWsamfs/T10K_cat
/dev/rmt/4cbn                  1001    ti      T10K    on
/dev/rmt/5cbn                  1002    ti      T10K    on
/dev/rmt/0cbn                  1003    ti      T10K    on
/dev/rmt/1cbn                  1004    ti      T10K    on
/dev/rmt/6cbn                  1005    ti      T10K    on
/dev/rmt/7cbn                  1006    ti      T10K    on
/dev/rmt/2cbn                  1007    ti      T10K    on
/dev/rmt/11cbn                 1008    ti      T10K    on
/dev/rmt/10cbn                 1009    ti      T10K    on
/dev/rmt/12cbn                 1010    ti      T10K    on

# Define Sun SAM-Remote server jimmy
/etc/opt/SUNWsamfs/rmt2000 2000 ss jimmy on
```

Example: Server Configuration File on Server ***jimmy***

```
# Server configuration file /etc/opt/SUNWsamfs/rmt2000 on jimmy.
# The eq of the automated library MUST match the eq of the
# automated library that you want to use in the mcf file.
tera
   media
     1000 ti 00002[0-9]
   endmedia
```

Configuration Files for Clients

The client must have Sun SAM-Remote configuration information in its `mcf` file and in its client configuration file. The following examples show these files.

Example: The ***mcf*** File on Client ***tera***

```
# mcf file for client (tera)
#

# Equipment                                        Eq  Eq Family Dev Additional
# Identifier                                       Ord Ty Set    St  Parameters
# ==========                                       === == ====== == ==========
samfs1                                             100 ms samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE5F00048F2Dd0s6 101 md samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE23000B24C2d0s6 102 md samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE50000ADAECd0s6 103 md samfs1 on

# Define a L500 with 2 drives
/dev/samst/c4t500104F0009C2F6Fu0 300 rb L500 on
/dev/rmt/0cbn                    301 li L500 on
/dev/rmt/1cbn                    302 li L500 on


# Define tera as a Sun SAM-Remote client using jimmy as the server
/etc/opt/SUNWsamfs/rmt2000       2000 ss jimmy on
/dev/samrd/rd0                   2001 rd jimmy on
/dev/samrd/rd1                   2002 rd jimmy on
/dev/samrd/rd2                   2003 rd jimmy on
/dev/samrd/rd3                   2004 rd jimmy on
/dev/samrd/rd4                   2005 rd jimmy on
/dev/samrd/rd5                   2006 rd jimmy on
/dev/samrd/rd6                   2007 rd jimmy on
/dev/samrd/rd7                   2008 rd jimmy on
```

Example: Client Configuration File on Client ***tera***

```
# cat /etc/opt/SUNWsamfs/rmt2000
# File /etc/opt/SUNWsamfs/rmt2000 on Sun SAM-Remote client tera: jimmy
```

How To Configure Recycling---Method 1

The procedure for configuring the recycling process includes a test for archiving and recycling. Because of the testing period, this procedure can take a day or two to complete, depending on how frequently files are archived and recycled.

> Note -
> Do not use the `chmed(1M)` command on the server to set the recycling flag (`+c`) for a client VSN. That action overrides the `no_recycle` list in the `/etc/opt/SUNWsamfs/recycler.cmd` file on the server.

Before starting the procedure, read About Recycling. Using the recycler in a Sun SAM-Remote environment requires a complete understanding of the steps in the recycling process. If you have not already familiarized yourself with the recycling process, do so now.

1. Make sure that the Sun SAM-Remote client and server are configured properly and that archiving is occurring.
   For more information, see Configuring the Sun SAM-Remote Software, which contains detailed information about configuring the Sun SAM-Remote client and server. That procedure includes steps for ensuring that archiving is taking place.
2. Edit the `archiver.cmd` file on the client system and add recycling directives.
   In this example, the recycling is performed by archive set, not by library. The directives specifying that recycling be done by archive set must appear in the `archiver.cmd` file.
   The example below shows the `archiver.cmd` file on client `zeke`. This file has been edited to communicate with the recycler.

   Example: ***archiver.cmd*** on a Client

```
# This is file /etc/opt/SUNWsamfs/archiver.cmd
# on Sun SAM-Remote client zeke.
#
archivemeta = off
archmax = li 12G

fs = samfs1
logfile = /var/adm/samfs1.archiver.log
no_archive tmp
all .
    1 -norelease 10m
    2 -norelease 10m

params
allsets -sort path -offline_copy stageahead -reserve set
allsets -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 2 -dataquantity 100G
allsets -ignore
allsets.1 -startage 8h -startsize 10G -drives 2
allsets.2 -startage 24h -startsize 20G -drives 2 -archmax 24G
endparams

vsns
all.1 li  ^10.*
all.2 li  ^20.*
endvsns
```

The directives shown in the example do the following:

- The `-recycle_hwm` directive sets the library's high-water mark for the archive set. When the utilization of the VSNs exceeds this percentage, recycling of the archive set begins.
- The `-recycle_ignore` directive is inserted only temporarily. This directive prevents recycling from occurring until you have configured and tested your environment. You can remove this directive in a later step.
- The `-recycle_mingain` directive is set high to ensure efficiency by limiting the amount of work needed to regain space.
- The `-recycle_vsncount 2` directive specifies that the recycler drain two VSN at a time. Do not let recycling overwhelm the system.

3. Edit the `recycler.cmd` file on the client and specify a log file to receive recycling log output.
   The example below shows the `recycler.cmd` file on client `zeke`, which has been edited to specify a recycler log file.

   Example: ***recycler.cmd*** on a Client

```
#
# This is the /etc/opt/SUNWsamfs/recycler.cmd file
# on client zeke.
#
logfile = /var/adm/recycler.log
```

4. Verify that the `archiver.cmd` file on the server is written to specify recycling by archive set.
   When using Sun SAM-Remote, you must specify that recycling be performed on an archive set basis, not by library. The directives specifying that recycling be done by archive set must appear in the `archiver.cmd` file.
   The following example shows the `archiver.cmd` file on server `sky`. This file specifies archiving by archive set.

   Example: ***archiver.cmd*** on a Server

```
# This is the archiver.cmd for the server (sky).
#
# Number of drives: 10
# Number of Mounted Filesystems: 1
# Number of Tests per Filesystem: 1
# Number of Archive Copies per Test: 2
#wait
#trace = /var/opt/SUNWsamfs/trace/archiver all
logfile = /var/opt/SUNWsamfs/log/archiver
interval = 1m
no_archive .
archmax = at 5G
drives = adic1 6
fs = samfs1
1 4h
testset testdir0
1 1m
2 1m
allsam1 .
1 1m
2 1m
params
allsam1.1 -drives 4 -drivemin 50m
allsam1.1 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
allsam1.1 -recycle_ignore
allsam1.2 -drives 4 -drivemin 50m
allsam1.2 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
allsam1.2 -recycle_ignore
testset.1 -drives 4 -drivemin 50m
testset.1 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
testset.1 -recycle_ignore
testset.2 -drives 4 -drivemin 50m
testset.2 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
testset.2 -recycle_ignore
endparams
vsns
samfs1.1 at 000000
allsam1.1 at 00000[1-5] # vsns 1 through 5.
allsam1.2 at 00000[6-9] # vsns 6 through 9.
testset.1 at 00001[0,4] # vsns 10 and 14.
testset.2 at 00001[5,9] # vsns 15 and 19.
endvsns
```

5. Edit the `recycler.cmd` file on the server. Modify the file to specify the following items:
    - A recycler log file to receive output from the recycler.
    - A `no_recycle` directive for the Sun SAM-Remote client's VSNs. The Sun SAM-Remote client is configured to write its copy 2 archive copies to cartridges in the Sun SAM-Remote server's library. The `no_recycle` directive is necessary to prevent the VSNs being used by the Sun SAM-Remote client for archiving from being recycled by the Sun SAM-Remote server.
      The following example shows the `recycler.cmd` file on server `sky`, which has been edited to specify a recycler log file.

    Example: ***`recycler.cmd`*** on a Server

    ```
    #
    # This is the /etc/opt/SUNWsamfs/recycler.cmd file
    # on Sun SAM-Remote server sky.
    #
    logfile = /var/opt/SUNWsamfs/recycler/recycler.log
    adic1 -ignore
    no_recycle at 00002[0-9] # Prevents VSNs assigned to zeke from
    # being recycled.
    ```

6. Use the `sam-recycler`(1M) command to test the recycler on the Sun SAM-Remote client.
   Run the recycler on the Sun SAM-Remote client system. This is a test to see if the recycler properly acknowledges the devices and VSNs specified in the configuration files.
   For example, you can use the following command to perform the initial test of the recycler:

```
zeke# sam-recycler -dvx
```

This testing is important, because if the recycler detects that the system on which it is running has no archive images on a particular VSN listed in any of that system's catalogs (including the historian catalog), the `recycler.sh` script can call for the cartridge to be labeled. Labeling a cartridge destroys all data on the cartridge. There is no communication between the Sun SAM-Remote client and the SAM-QFS servers to inform each side of the presence of archive copies. All such information is provided locally from local SAM-QFS file systems.
The recycler runs and logs its activity to the recycler log file. The recycler log file is defined in the `recycler.cmd` file. For more information about the `sam-recycler`(1M) command, see the `sam-recycler`(1M) man page.

7. Examine the recycler log file to find the following message:

```
Recycling is ignored on this archive set.
```

The following example shows a sample log file.

Example: Recycler Log File on a Client

```
# recycler.log from client zeke.
========== Recycler begins at Mon Jun 4 09:49:41 2001 ===========
Initial 7 catalogs:
0 Family: stk_l20 Path: /var/opt/SUNWsamfs/catalog/L20_cat
Vendor: STK Product: L20
SLOT ty capacity space vsn
0 lt 33.0G 33.0G 000173
1 lt 32.8G 44.1M CEL170
2 lt 33.0G 33.0G CEL139
4 lt 32.8G 16.8G CFC504
5 lt 33.0G 33.0G CFC503
6 lt 32.9G 0 CSM689
7 lt 32.9G 19.6G CSM690
8 lt 33.0G 33.0G CSM691
9 lt 33.0G 33.0G CSM692
10 lt 10.0G 10.0G CLN018
11 lt 33.0G 33.0G 000766
Total Capacity: 339.2G bytes, Total Space Available: 244.3G bytes
Volume utilization 27%, high 95% VSN_min 50%
Recycling is ignored on this robot.

1 Family: skyrs Path: /var/opt/SUNWsamfs/catalog/sky_cat
Vendor: (NULL) Product: (NULL)
SLOT ty capacity space vsn
0 at 48.5G 23.3G 000020
1 at 23.8G 23.8G 000021
2 at 48.5G 48.5G 000022
3 at 48.5G 48.5G 000023
4 at 48.5G 48.5G 000024
5 at 48.5G 2.6G 000025
6 at 48.5G 361.4k 000026
7 at 48.5G 48.5G 000027
8 at 48.5G 48.5G 000028
9 at 48.5G 0 000029
Total Capacity: 460.8G bytes, Total Space Available: 292.5G bytes
Volume utilization 36%, high 95% VSN_min 50%
Recycling is ignored on this robot.

2 Family: hy Path: /var/opt/SUNWsamfs/catalog/historian
Vendor: Sun SAM-FS Product: Historian
SLOT ty capacity space vsn
(no VSNs in this media changer)
Total Capacity: 0 bytes, Total Space Available: 0 bytes
Volume utilization 0%, high 95% VSN_min 50%
Recycling is ignored on this robot.
```

```
3 Family: defaultset.1 Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS Product: Archive set
SLOT ty capacity space vsn
0 lt 33.0G 33.0G 000766
1 lt 33.0G 33.0G 000173
2 lt 32.9G 0 CSM689
3 lt 32.9G 19.6G CSM690
4 lt 33.0G 33.0G CSM691
5 lt 33.0G 33.0G CSM692
Total Capacity: 197.6G bytes, Total Space Available: 151.5G bytes
Volume utilization 23%, high 60% VSN_min 90%
Recycling is ignored on this archive set.

4 Family: defaultset.2 Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS Product: Archive set
SLOT ty capacity space vsn
0 lt 32.9G 0 CSM689
1 at 48.5G 23.3G 000020
2 at 23.8G 23.8G 000021
3 at 48.5G 2.6G 000025
4 at 48.5G 361.4k 000026
5 at 48.5G 48.5G 000027
6 at 48.5G 48.5G 000028
7 at 48.5G 0 000029
Total Capacity: 348.0G bytes, Total Space Available: 146.8G bytes
Volume utilization 57%, high 60% VSN_min 90%
Recycling is ignored on this archive set.

5 Family: archiveset.1 Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS Product: Archive set
SLOT ty capacity space vsn
0 lt 32.8G 44.1M CEL170
1 lt 32.8G 16.8G CFC504
2 lt 33.0G 33.0G CFC503
Total Capacity: 98.6G bytes, Total Space Available: 49.8G bytes
Volume utilization 49%, high 60% VSN_min 90%
Recycling is ignored on this archive set.

6 Family: archiveset.2 Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS Product: Archive set
SLOT ty capacity space vsn
0 at 48.5G 23.3G 000020
1 at 23.8G 23.8G 000021
2 at 48.5G 48.5G 000022
3 at 48.5G 48.5G 000023
4 at 48.5G 48.5G 000024
Total Capacity: 218.0G bytes, Total Space Available: 192.8G bytes
Volume utilization 11%, high 60% VSN_min 90%
Recycling is ignored on this archive set.

21 VSNs:

---Archives--- -----Percent----- defaultset.1
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
in multiple sets 0 0 0 100 0 stk_l20:lt:CSM689
partially full 111 2.8G 8 31 61 stk_l20:lt:CSM690
empty VSN 0 0 0 0 100 stk_l20:lt:000173
empty VSN 0 0 0 0 100 stk_l20:lt:CSM691
empty VSN 0 0 0 0 100 stk_l20:lt:CSM692
empty VSN 0 0 0 0 100 stk_l20:lt:000766

---Archives--- -----Percent----- defaultset.2
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 skyrs:at:000029
no-data VSN 0 0 0 99 1 skyrs:at:000026
partially full 111 2.8G 6 88 6 skyrs:at:000025
empty VSN 0 0 0 0 100 skyrs:at:000028
empty VSN 0 0 0 0 100 skyrs:at:000027

---Archives--- -----Percent----- archiveset.1
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 99 1 stk_l20:lt:CEL170
partially full 677 2.3G 8 40 52 stk_l20:lt:CFC504
empty VSN 0 0 0 0 100 stk_l20:lt:CFC503
```

```
---Archives--- -----Percent----- archiveset.2
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
in multiple sets 0 0 0 51 49 skyrs:at:000020
empty VSN 0 0 0 0 100 skyrs:at:000022
empty VSN 0 0 0 0 100 skyrs:at:000023
empty VSN 0 0 0 0 100 skyrs:at:000024
in multiple sets 0 0 0 0 100 skyrs:at:000021

---Archives--- -----Percent----- stk_l20
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
empty VSN 0 0 0 0 100 stk_l20:lt:CLN018
partially full 13 80.3k 0 0 100 stk_l20:lt:CEL139
```

```
Recycler finished.
========== Recycler ends at Mon Jun 4 09:49:53 2001 ===========
```

8. Issue the `sam-recycler`(1M) command from the Sun SAM-Remote server to verify that the recycler is not recycling any VSNs reserved for the Sun SAM-Remote client.
   For example:

```
sky# sam-recycler -dvx
```

The preceding command runs the recycler and writes its activity to the recycler log file. For more information about the `sam-recycler`(1M) command, see the `sam-recycler`(1M) man page.
The following example shows a sample recycler log file.

Example: Recycler Log File on a Server

```
# recycler.log file from server sky.
========== Recycler begins at Mon Jun 4 09:50:44 2001 ===========
Initial 6 catalogs:
0 Family: adic1 Path: /var/opt/SUNWsamfs/catalog/adic1
Vendor: ADIC Product: Scalar 1000
SLOT ty capacity space vsn
0 at 1.3G 1.2G 000001
1 at 1.3G 1.3G 000002
2 at 1.3G 1.3G 000004
3 at 48.5G 0 000010
4 at 48.5G 0 000011
5 at 48.5G 43.5G 000018
6 at 48.5G 0 000019
7 at 48.5G 23.3G 000020
8 at 23.8G 23.8G 000021
9 at 48.5G 48.5G 000022
10 at 48.5G 48.5G 000023
11 at 48.5G 48.5G 000024
12 at 48.5G 2.6G 000025
13 at 48.5G 361.4k 000026
14 at 48.5G 48.5G 000027
15 at 48.5G 48.5G 000028
16 at 48.5G 0 000029
17 at 1.3G 1.3G 000005
18 at 48.5G 48.5G 000016
19 at 23.8G 23.8G CLN001
20 at 23.8G 23.8G CLN002
21 at 23.8G 23.8G CLN004
22 at 23.8G 23.8G CLN003
23 at 48.5G 421.6M 000015
24 at 1.3G 1.3G 000000
25 at 48.5G 0 000013
26 at 1.3G 1.3G 000003
27 at 48.5G 43.6G 000007
28 at 48.5G 41.8G 000008
29 at 48.5G 46.9G 000006
30 at 48.5G 48.3G 000009
31 at 48.5G 0 000014
32 at 48.5G 0 000012
33 at 48.5G 40.1G 000017
Total Capacity: 1.2T bytes, Total Space Available: 708.7G bytes
Volume utilization 43%, high 95% VSN_min 50%
Recycling is ignored on this robot.

1 Family: hy Path: /var/opt/SUNWsamfs/catalog/historian
Vendor: Sun SAM-FS Product: Historian
SLOT ty capacity space vsn
(no VSNs in this media changer)
Total Capacity: 0 bytes, Total Space Available: 0 bytes
Volume utilization 0%, high 95% VSN_min 50%
Recycling is ignored on this robot.
```

```
2 Family: testset.1 Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS Product: Archive set
SLOT ty capacity space vsn
0 at 48.5G 0 000010
1 at 48.5G 0 000014
Total Capacity: 97.1G bytes, Total Space Available: 0 bytes
Volume utilization 100%, high 60% VSN_min 90%: *** Needs recycling ***
Recycling is ignored on this archive set.

3 Family: testset.2 Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS Product: Archive set
SLOT ty capacity space vsn
0 at 48.5G 0 000019
1 at 48.5G 421.6M 000015
Total Capacity: 97.1G bytes, Total Space Available: 421.6M bytes
Volume utilization 99%, high 60% VSN_min 90%: *** Needs recycling ***
Recycling is ignored on this archive set.

4 Family: allsam1.1 Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS Product: Archive set
SLOT ty capacity space vsn
0 at 1.3G 1.2G 000001
1 at 1.3G 1.3G 000002
2 at 1.3G 1.3G 000004
3 at 1.3G 1.3G 000005
4 at 1.3G 1.3G 000003
Total Capacity: 6.5G bytes, Total Space Available: 6.3G bytes
Volume utilization 3%, high 60% VSN_min 90%
Recycling is ignored on this archive set.

5 Family: allsam1.2 Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS Product: Archive set
SLOT ty capacity space vsn
0 at 48.5G 43.6G 000007
1 at 48.5G 41.8G 000008
2 at 48.5G 46.9G 000006
3 at 48.5G 48.3G 000009
Total Capacity: 194.2G bytes, Total Space Available: 180.6G bytes
Volume utilization 6%, high 60% VSN_min 90%
Recycling is ignored on this archive set.

Need to select candidate for media changer testset.1 to free up 39.8G bytes.
Quantity of data to move limited to (no limit) bytes and 1 VSNs.
Checking 000010. Need to free 39.8G, quantity limit: (no limit), VSN count: 1.
VSN is in correct media changer... good.
VSN is not already recycling... good.
VSN has no request files... good.
VSN has no 'archive -n' files...good.
VSN was not specified as "no_recycle" in recycler.cmd file... good.
VSN does not exceed VSN count limit... good.
VSN does not exceed data quantity limit... good.
VSN meets minimum gain requirement.
Recycling is ignored on this media changer - VSN not marked for recycling.

Checking 000014. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN is in correct media changer... good.
VSN is not already recycling... good.
VSN has no request files... good.
VSN has no 'archive -n' files...good.
VSN was not specified as "no_recycle" in recycler.cmd file... good.
VSN exceeds VSN count limit - skipped.
Checking 000019. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000015. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking 000001. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000003. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000004. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000005. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
```

```
Checking 000002. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000008. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000007. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking 000006. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000009. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000011. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000029. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000013. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000012. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking 000026. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000025. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000020. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000017. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000018. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN003. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking 000021. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000022. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000027. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000028. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000023. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000024. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000016. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking CLN001. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN002. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN004. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000000. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
No candidate was found in this media changer.

Need to select candidate for media changer testset.2 to free up 38.8G bytes.
Quantity of data to move limited to (no limit) bytes and 1 VSNs.
Checking 000010. Need to free 38.8G, quantity limit: (no limit), VSN count: 1.
VSN not in correct media changer.
Checking 000014. Need to free 38.8G, quantity limit: (no limit), VSN count: 1.
VSN not in correct media changer.

Checking 000019. Need to free 38.8G, quantity limit: (no limit), VSN count: 1.
VSN is in correct media changer... good.
VSN is not already recycling... good.
VSN has no request files... good.
VSN has no 'archive -n' files...good.
VSN was not specified as "no_recycle" in recycler.cmd file... good.
VSN does not exceed VSN count limit... good.
VSN does not exceed data quantity limit... good.
VSN meets minimum gain requirement.
Recycling is ignored on this media changer - VSN not marked for recycling.
```

```
Checking 000015. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN is in correct media changer... good.
VSN is not already recycling... good.
VSN has no request files... good.
VSN has no 'archive -n' files...good.
VSN was not specified as "no_recycle" in recycler.cmd file... good.
VSN exceeds VSN count limit - skipped.

Checking 000001. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000003. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000004. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000005. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000002. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking 000008. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000007. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000006. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000009. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000011. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000029. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking 000013. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000012. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000026. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000025. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000020. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000017. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking 000018. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN003. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000021. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000022. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000027. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000028. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000023. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking 000024. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000016. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN001. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN002. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN004. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000000. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
No candidate was found in this media changer.
34 VSNs:
```

```
---Archives--- -----Percent----- testset.1
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 adic1:at:000010
no-data VSN 0 0 0 100 0 adic1:at:000014

---Archives--- -----Percent----- testset.2
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 adic1:at:000019
partially full 677 2.3G 5 93 2 adic1:at:000015

---Archives--- -----Percent----- allsam1.1
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
partially full 97 173.8M 1 9 90 adic1:at:000001
no-data VSN 0 0 0 2 98 adic1:at:000003
no-data VSN 0 0 0 2 98 adic1:at:000004
empty VSN 0 0 0 0 100 adic1:at:000005
empty VSN 0 0 0 0 100 adic1:at:000002

---Archives--- -----Percent----- allsam1.2
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 13 87 adic1:at:000008
partially full 98 1.6G 3 7 90 adic1:at:000007
no-data VSN 0 0 0 3 97 adic1:at:000006
empty VSN 0 0 0 0 100 adic1:at:000009

---Archives--- -----Percent----- adic1
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 adic1:at:000011
no_recycle VSN 0 0 0 100 0 adic1:at:000029
no-data VSN 0 0 0 100 0 adic1:at:000013
no-data VSN 0 0 0 100 0 adic1:at:000012
no_recycle VSN 0 0 0 99 1 adic1:at:000026
no_recycle VSN 0 0 0 94 6 adic1:at:000025
no_recycle VSN 0 0 0 51 49 adic1:at:000020
no-data VSN 0 0 0 17 83 adic1:at:000017
no-data VSN 0 0 0 10 90 adic1:at:000018
empty VSN 0 0 0 0 100 adic1:at:CLN003

no_recycle VSN 0 0 0 0 100 adic1:at:000021
no_recycle VSN 0 0 0 0 100 adic1:at:000022
no_recycle VSN 0 0 0 0 100 adic1:at:000027
no_recycle VSN 0 0 0 0 100 adic1:at:000028
no_recycle VSN 0 0 0 0 100 adic1:at:000023
no_recycle VSN 0 0 0 0 100 adic1:at:000024
empty VSN 0 0 0 0 100 adic1:at:000016
empty VSN 0 0 0 0 100 adic1:at:CLN001
empty VSN 0 0 0 0 100 adic1:at:CLN002
empty VSN 0 0 0 0 100 adic1:at:CLN004
partially full 12 88.3k 0 0 100 adic1:at:000000
```

```
Recycler finished.
========== Recycler ends at Mon Jun 4 09:51:05 2001 ===========
```

9. Analyze the server and client `recycler.log` files to choose VSNs that are candidates for recycling.
   Near the end of the `recycler.log` file is a `Status` column.
   - In the client log files, VSNs with the following types of status entries are candidates for recycling:
     - `no-data VSN`. To recycle a `no-data` VSN, see To Recycle `no-data` VSNs.
     - `partially full`. To recycle a `partially full` VSN, see To Recycle `partially full` VSNs.
   - In the server log file, the best candidates for recycling are those with a 0 value in the `Count`, `Bytes`, and `Use` columns.

How To Recycle **`no-data`** VSNs

The `no-data` VSNs are the easiest VSNs to recycle. For these, the `Count`, `Bytes`, and `Use` field values are all 0 (zero).

1. Examine the `recycler.log` file from the client to see if there are any `no-data` VSNs.
   VSNs 000029 and 000026 from the client `zeke` can be considered for recycling because they are `no-data` VSNs, as shown in the code below.

```
# From the client zeke recycler.log file:
---Archives--- -----Percent----- defaultset.2
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 skyrs:at:000029
no-data VSN 0 0 0 99 1 skyrs:at:000026
partially full 111 2.8G 6 88 6 skyrs:at:000025
empty VSN 0 0 0 0 100 skyrs:at:000028
empty VSN 0 0 0 0 100 skyrs:at:000027
```

2. Examine the `recycler.log` file from the server and determine if the VSNs you selected from the previous step are represented identically in the server's recycler log file.
   Verify that there is no active data from the server archived on those VSNs.
   The code below shows the data for the `no_recycle` VSNs in the server's `recycler.log` file. For VSNs 000029 and 000026, the data in the server's `recycler.log` file is identical to that in the client's `recycler.log` file.

```
# From the Server log file:
---Archives--- -----Percent----- adic1
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 adic1:at:000011
no_recycle VSN 0 0 0 100 0 adic1:at:000029
no-data VSN 0 0 0 100 0 adic1:at:000013
no-data VSN 0 0 0 100 0 adic1:at:000012
no_recycle VSN 0 0 0 99 1 adic1:at:000026
no_recycle VSN 0 0 0 94 6 adic1:at:000025
no_recycle VSN 0 0 0 51 49 adic1:at:000020
no-data VSN 0 0 0 17 83 adic1:at:000017
no-data VSN 0 0 0 10 90 adic1:at:000018
empty VSN 0 0 0 0 100 adic1:at:CLN003
.
.
.
```

3. If no active data from the server is archived on the selected VSNs, use the `tplabel`(1M) or `odlabel`(1M) command to relabel the VSNs.

   > Note -
   > This destroys all data on the VSN and reclaims space.

   For example, for tape VSN 000029, use the following command:

```
sky# tplabel -vsn 000029 -old 000029 at.000029
```

When this VSN 000029 is relabeled, you regain 100 percent of the space on that VSN.
If the media is a magneto-optical disk, use the odlabel(1M) command. For more information, see the odlabel(1M) man page.

How To Recycle **partially full** VSNs

The VSNs for which a partially full status is reported can also be recycled.

1. Examine the recycler.log file from the client to see if there are any partially full VSNs.
   You can consider VSN 000025 from the client, zeke, for recycling because its status is partially full, as shown below.

```
# From the client zeke recycler.log file:
---Archives--- -----Percent----- defaultset.2
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 skyrs:at:000029
no-data VSN 0 0 0 99 1 skyrs:at:000026
partially full 111 2.8G 6 88 6 skyrs:at:000025
empty VSN 0 0 0 0 100 skyrs:at:000028
empty VSN 0 0 0 0 100 skyrs:at:000027
```

   VSN 000025 shows that 6 percent of its space is in use. These are active archive images that must be rearchived before this VSN can be recycled. The following steps in this process show how to ensure that these active archive images are rearchived to another VSN.

2. Examine the recycler.log file from the server side to ensure that no active data from the server is archived on that VSN.
   The server's recycler.log file indicates that VSN 000025 is 6 percent free, which is the same percentage that was reported in the client's recycler.log file. The server is not aware of the client's archive images, so it reports that all of the remaining 94 percent is consumed by obsolete archive images.

```
# From the Server log file:
---Archives--- -----Percent----- adic1
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 adic1:at:000011
no_recycle VSN 0 0 0 100 0 adic1:at:000029
no-data VSN 0 0 0 100 0 adic1:at:000013
no-data VSN 0 0 0 100 0 adic1:at:000012
no_recycle VSN 0 0 0 99 1 adic1:at:000026
no_recycle VSN 0 0 0 94 6 adic1:at:000025
no_recycle VSN 0 0 0 51 49 adic1:at:000020
no-data VSN 0 0 0 17 83 adic1:at:000017
.
.
.
```

3. On the server, use the chmed(1M) command with the +c option to rearchive the active files on the VSN.

```
sky# chmed +c at.000025
```

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

4. On the client, use the sam-recycler(1M) command to run the recycler again.

```
zeke# sam-recycler -dvx
```

   This marks each active file to be rearchived to another VSN.

5. Start the archiver.
   You can do this either by letting the archiver run normally, or by typing :arrun from the samu(1M) utility on the client. For more information about the :arrun command, see the samu(1M) man page.
6. When archiving is complete, issue the sam-recycler(1M) command to rerun the recycler on the client.

```
zeke# sam-recycler -dvx
```

This ensures that all active files have been rearchived.

7. If the Count, Bytes, and Use field values are all 0 (zero), use the `tplabel`(1M) or `odlabel`(1M) command to relabel the VSN from the server.
   For example, for tape VSN 000025, use the following command:

```
sky# tplabel -vsn 000025 -old 000025 at.000025
```

   This command relabels the VSN and destroys all data on it. After this VSN is relabeled, you regain 88 percent of the space on this VSN.
   If the media had been a magneto-optical disk, you would have used the `odlabel`(1M) command. For more information about the `odlabel`(1M) command, see the `odlabel`(1M) man page.

Recycling in a Sun SAM-Remote Environment---Method 2

This section presents another way you can recycle volumes using Sun SAM-remote software.

> Caution -
> Use the recycler in a Sun SAM-Remote environment only after following the steps in this procedure completely and only after testing your configuration to verify that recycling is taking place correctly.

How To Configure Recycling---Method 2

1. On the Sun SAM-Remote client, issue the `sam-recycler`(1M) command to determine which volumes are the best candidates for recycling. For example:

```
client# sam-recycler -dvx
```

2. Analyze the recycler log file for recycling candidates.
   Toward the end of the `recycler.log` file is a `Status` column. VSNs with the following types of status entries in the client log file are candidates for recycling:
   - `no-data VSN`. To recycle a `no-data` VSN, see To Recycle `no-data` VSNs.
   - `partially full`. To recycle a `partially full` VSN, see To Recycle `partially full` VSNs.
     In the server log file, the best candidates for recycling are those with a 0 value in the `Count`, `Bytes`, and `Use` columns.
3. On the Sun SAM-Remote server, issue the `chmed`(1M) command to set the recycle flag on the selected VSNs.
   For example:

```
server# chmed +c at.00025
```

4. Wait until the VSNs being recycled are drained completely of archive images.
   The archiver on the client side does this.
5. On the Sun SAM-Remote server, issue the `tplabel`(1M) or `odlabel`(1M) command, depending on the archive media, to relabel the volumes.
6. On the Sun SAM-Remote server, clear any flags, such as `R` or `c` that prevent the volumes from being used for archiving on the Sun SAM-Remote client.

End of Sun Storage Archive Manager (SAM-QFS) Configuration and Administration Guide

About Archiving

Contents

About Archiving

Archiving is the process of copying a file from a file system to a volume that resides on a removable media cartridge or on a disk partition of another file system. When using the Sun Storage Archive Manager (SAM), you can specify that files be archived immediately, specify that files never be archived, and perform other tasks.

The term archive media is used to refer to the various cartridges or disk slices to which archive volumes are written. This page provides general guidelines for developing archive policies for your site and describes the archiver's theory of operations. See Configuring the Archiver for details on how to configure the archiver.

Planning Archiving Operations

The archiver automates storage management operations using the `archiver.cmd` file. Before writing this file, review the following general guidelines. They can improve the performance of your file system and the archiver, which can help ensure that your data is stored in the safest way possible.

- Save your archive logs. The archive logs are essential to recovering data, even if the SAM software is unavailable. Keep these logs in a safe place in the event of a catastrophic disaster.
- Use regular expressions for volume ranges. Enable the system to put files on many different volumes. Volume ranges allow the system to run continuously. If you use specific volume names, the archive set copies can fill a volume rapidly, causing workflow problems as you replace media.
- Set an optimal archive interval. The archive interval is the time between file system scans. Set your archive interval based on how often files are created and modified, and whether you want to save all modified copies. An archive interval that is too short keeps the archiver scanning almost continuously.
- Consider the number of file systems you are using. Multiple file systems increase the performance of the archiver over a single file system. Multiple file systems can be scanned in less time than a single file system.
- Use directory structures to organize your files in a file system. To maximize performance, do not place more than 10,000 files in a directory.
- Always make a minimum of two file copies on two separate volumes. Putting data on a single media type puts your data at risk if physical problems with the media occur.
- Back up your metadata (directory structure, file names, and so on) Use `samfsdump`(1M) on a regular basis. The metadata is stored in an archive set that has the same name as the file system. You can use this information to recover a file system in the event of a disaster. If you do not want the archiver to back up your metadata, you can set `archivemeta=off` in the `archiver.cmd` file and schedule the `samfsdump` command to run in a `cron` file.

Preview Queue

The archiver and stager processes both can request that media be loaded and unloaded. If the number of requests exceeds the number of drives available for media loads, the excess requests are sent to the preview queue.

By default, preview requests are satisfied in first-in-first-out (FIFO) order. You can override the FIFO default by entering directives in the preview command file, which is written to `/etc/opt/SUNWsamfs/preview.cmd`. For more information about this file and setting priorities for archiving and staging, see Prioritizing Preview Requests.

Archiver Daemons

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

When Sun Storage Archive Manager (SAM) is initiated, its `sam-fsd` daemon starts the `sam-archiverd` daemon.
The `sam-archiverd` daemon executes the `archiver`(1M) command to read the `archiver.cmd` file and builds the tables necessary to control archiving.

The `sam-archiverd` daemon starts a `sam-arfind` process for each mounted file system. When a file system is unmounted, it stops the associated `sam-arfind` process.

The `sam-archiverd` process monitors `sam-arfind` and processes signals from an operator or other processes.

Archiving Operations

Operator actions are not required to archive files. The archiver writes files to a volume on the archive media. Archive media can contain one or more volumes. Each volume is identified by a unique identifier called a volume serial name (VSN).

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

To be a candidate for archiving or re-archiving, the data in a file must change. A file is not archived if it is accessed but not changed. For example, issuing a `touch`(1) command on a file does not cause it to be archived or re-archived. If the `mv`(1) command is used to rename a file, the file might move to a different archive set. In this case, the archiver software re-evaluates the archive copies to determine whether the file needs to be archived or re-archived.

> Note
> Because issuing an `mv`(1) command alters the file name but not the file data, this action can have ramifications for disaster recovery if you plan to restore from `tar`(1) files. For more information about disaster recovery, see SAM-QFS Troubleshooting.

The archive operation is affected by the following factors:

- archive age The period of time that has passed since the file was last modified. The archive age can be defined for each archive copy. Use the `touch`(1) command to change the default time references on their files to values in the past or future. However, this practice can cause unexpected archiving results. To avoid problems, the archiver adjusts the references so that they are always at a point between the file creation time and the present time.
- archive priority This value is computed from file property characteristics and from file property multipliers associated with the archive set, as follows:

```
<archive-priority> = <file-property-value> x <property-multiplier>
```

  - Most file-property-value numbers are 1 (for true) or 0 (for false). For instance, the value of the property copy 1 is 1 if archive copy 1 is being made. The values of copy 2, copy 3, and copy 4 are therefore 0. Other properties, such as archive age and file size, can have values other than 0 or 1.
  - The property-multiplier value is determined from the `-priority` parameters for the archive set. Various aspects of a file, such as age or size, can be given values to determine the archive request's priority. The default value for all property multipliers is 0.0. For more information about the `-priority` parameter, see the `archiver.cmd`(4) man page.

The archive-priority and the property-multiplier values are floating-point numbers.

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

- Step 1 Identifying Files to Archive
- Step 2 Composing Archive Requests
- Step 3 Scheduling Archive Requests
- Step 4 Archiving the Files in an Archive Request
- Sample Default Output from archiver(1M) -l Command

Step 1 Identifying Files to Archive

A separate `sam-arfind` process monitors each mounted file system to determine the files that need archiving. The file system notifies this `sam-arfind` process whenever a file is changed in a manner that affects its archival state, such as file modification, re-archiving, unarchiving, and renaming.
The `sam-arfind` process examines the file to determine which archive set the file belongs to and what action to take.

To determine a file's archive set,the `sam-arfind` process uses the following file properties descriptions:

- The directory path portion of the file's name and, optionally, the complete file name using a regular expression
- The user name of the file's owner
- The group name of the file's owner
- A minimum file size

- A maximum file size

To determine the archive action: the sam-arfind process performs the following evalutation:

- If the archive age of the file for one or more copies has been met or exceeded, the sam-arfind process adds the file to one or more archive requests for the archive set. An archive request is a collection of files that belong to the same archive set. The archive request resides in the /var/opt/SUNWsamfs/archiver/file_sys/ArchReq directory. The files in this directory are binary files. To display them, use the showqueue(1M) command. Separate archive requests are used for files not yet archived and for files being re-archived. This allows scheduling to be controlled independently for these two types of files.

- If the archive age of the file for one or more copies has not been met, the directory in which the file resides and the time at which the archive age is reached is added to a scan list. Directories are scanned as the scan list times are reached. Files that have reached their archive age are added to archive requests.

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

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

The archive action is accomplished using two methods:

- Continuous Archiving
- Scanned Archiving

Continuous Archiving

With the continuous archiving method, the archiver works with the file system to determine which files need to be archived.
Continuous archiving is the default archiving method (the archiver.cmd file parameter is examine=noscan) and operates with the following default start conditions:

- The archive starts every two hours.
- The archive waits until there is at least 90% of the archmax value of data to archive.
- The archive waits until there is at least 500,000 files to archive.
  When any of the scheduling start conditions is reached, the sam-arfind process sends each archive request to the archiver daemon, sam-archiverd, to set the schedule for file copying to archive media.

To control the schedule of the archive operation, specify the start conditions for each archive set by using the -startage, -startcount, and -startsize parameters. These conditions enable you to optimize archive timeliness versus archive work done. For example:

- If creating files that you want archived together takes an hour, set the -startage parameter to one hour (-startage 1h) to ensure that all files are created before the archive request occurs.
- If you want all archive requests to be no less than 150 gigabytes of data, specify a -startsize of 150 gigabytes (-startsize 150g) to direct the archiver to wait until 150 gigabytes of data are ready to be archived.
- If you know that 3000 files are available for archival, specify -startcount 3000 to ensure that the files are archived together.

For more information about archiving parameters, see Global Archiving Directives.

Scanned Archiving

With the scanning method, the archiver checks the file systems periodically and selects files for archiving. To enable the scanning method and to disable continuous scanning, specify examine=scan in the archiver.cmd file.

The sam-arfind process scans each file system periodically to determine which files need archiving and to place them in archive requests. The first scan is a directory scan, in which sam-arfind descends recursively through the directory tree. The process examines each file and sets the file status flag to archdone if the file does not need archiving. During successive scans, the sam-arfind process scans the .inodes file and examines only inodes without the archdone flag.

When the scan is complete, the sam-arfind process sends each archive request to the archiver daemon, sam-archiverd, to be scheduled for file copying to archive media. The sam-arfind process then sleeps for the duration specified by the interval=time directive. At the end of the interval, the sam-arfind process resumes scanning.

For information about controlling the setting of the archdone flag, see The setarchdone Directive: Controlling the Setting of the archdone Flag.

Step 2 Composing Archive Requests

Using either archiving methods, the result is that the `sam-arfind` process sends each archive request to the archiver daemon, `sam-archiverd` for composing. This section describes the composing process.

Composing is the process of selecting the files from the archive request to be archived at one time. Because of the capacity of the archive media or of the controls specified in the archiver command file, the files in an archive request might not be archived all at the same time. When the archive copy operation is complete for an archive request, the archive request is recomposed if files remain to be archived.

The `sam-archiverd` daemon places the files in the archive request according to certain default and site-specific criteria. The default operation is to archive all the files in an archive request by their full path name so that files in the same directories are also together on the archive media. The site-specific criteria enable you to control the order in which files are archived and how they can be distributed on volumes. These criteria, called archive set parameters, are evaluated in the following order: `-reserve`, `-sort`, `-rsort` (reverse sort), and `-drives`. For more information, see the `archiver.cmd`(4) man page.

When the `-reserve` owner parameter is specified, the `sam-archiverd` daemon orders the files in the archive request according to the file's directory path, user name, or group name. The files belonging to the owner are selected for archiving. The remaining files are archived later.

When the `-sort` or `-rsort` method parameter is specified, the `sam-archiverd` daemon orders the files according to the specified sort method such as age, size, or directory location.

When an archive request contains both online and offline files, the online files are selected for archiving first.
In the absence of a specified sort method, the offline files are ordered by the volume on which the archive copies reside. This rule ensures that all files in each archive set on the same volume are staged at the same time in the order in which they were stored on the media. When more than one archive copy of an offline file is being made, the offline file is not released until all required copies are made. All the files to be staged from the same volume as the first file are selected for archiving.

> Note
> The `-sort` and `-rsort` parameters can have a negative effect on performance during archiving of offline files if the order of the files to be archived does not match the order of the volumes needed for the offline files. Use these parameters only for the first archive copy to be made. Other copies must maintain the order of the first copy if enough archive media space is available when the copies are started.

After being composed, the archive requests are entered in the `sam-archiverd` daemon's scheduling queue, as described in the next section.

Step 3 Scheduling Archive Requests

The scheduler in the `sam-archiverd` daemon executes on demand when one of the following conditions exists:

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

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

The following must be true in order for archive requests to be scheduled:

- Drives must be available for making file copies.
- Volumes must be available and have sufficient space to hold the files in the archive request.

Drives

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

If the total size of files in the archive request is less than the `-drivemin` value, one drive is used. The `-drivemin` value is either the value specified by the `-drivemin` parameter or the `archmax` value. The `archmax` value is specified by the `-archmax` parameter or the value defined for the media. For more information about the `-archmax` parameter and the `archmax=` directive, see the `archiver.cmd`(4) man page.

If the total size of files in the archive request is more than the `-drivemin` value, the number of drives used is determined by the total size of the files divided by the `-drivemin` value. If the number of drives available is fewer than that specified by the `-drives` parameter, the smaller number of drives is used.

Drives can take varying amounts of time to archive files. The `-drivemax` parameter specifies the maximum number of bytes to be written to a drive before that drive is rescheduled for more data. You can use the `-drivemax` parameter to obtain better drive utilization.

Volumes

For archiving to occur, at least one volume must have enough space to hold at least some of the files in the archive request. The volume that has most recently been used for the archive set is the one scheduled, if it has enough space. This volume must not be in use by the archiver.

If a volume usable for the archive set is busy, another is selected, unless the `-fillvsns` parameter is specified. In this case, the archive request cannot be scheduled.

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

You can specify volume overflow for the archive set by using the `-ovflmin` parameter, or for the media by using the `ovflmin=` directive. For more information about the `-ovflmin` parameter and the `ovflmin=` directive, see the `archiver.cmd`(4) man page. The `ovflmin` specification determines the file size threshold above which additional volumes or media are assigned for archiving. An `ovflmin` value specified for the archive set takes precedence over an `ovflmin` value specified for the media.

If the size of the files is less than the value of `ovflmin`, the files cannot be archived. An appropriate message for this condition is sent to the log. If the size of the files is more than the value of `ovflmin`, additional volumes are assigned as required. Volumes are selected in order of decreasing size in order to minimize the number of volumes required. If no usable volumes can be found for the archive request, the archive request waits until volumes become available.

Certain properties, such as whether the file is online or offline, are used in conjunction with the archive priority to determine the scheduling priority for a particular archive request. For more information about customizing the priority multiplier, see the `-priority` parameters described on the `archiver.cmd`(4) man page.

For each archive request, the `sam-archiverd` daemon computes the scheduling priority by adding the archive priority to multipliers associated with various system resource properties. These properties are associated with the number of seconds for which the archive request has been queued, whether the first volume to be used in the archiving process is loaded into a drive, and so on.

Using the adjusted priorities, the `sam-archiverd` daemon assigns each ready archive request to be copied, as described in the next section.

Step 4 Archiving the Files in an Archive Request

When an archive request is ready to be archived, the `sam-archiverd` daemon marks the archive file boundaries so that each archive file's size is less than the specified `-archmax` value. If a single file is larger than this value, it becomes the only file in an archive file.

For each archive request and each drive to be used, the `sam-archiverd` daemon assigns the archive request to a `sam-arcopy` process to copy the files to the archive media. The archive information is entered into the inode.

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

For each file that was staged, the disk space is released until all files in the list have been archived.

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

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

You can also specify timeouts for archiving operations in the `archiver.cmd` file. The directive is as follows:

```
timeout = [<operation> | <media>] <time>
```

operation is one of the following:

- `read`- Reading the file from the disk. The default timeout is 1 minute.
- `request`- Requesting the archive media. The default timeout is 15 minutes.
- `stage`- Staging the file to be archived. The default timeout is 0 minutes.
- `write`- Writing to the archive media. The default timeout is 15 minutes.

The timeout value for the write operation can also be specified for individual media.

Notification of Schedule Queue Time

You can configure the `-queue_time_limit` time archive set parameter in the `archiver.cmd` file to notify the administrator when an archive request has been in the schedule queue longer than a certain amount of time. If the archive request remains in the queue at the end of this time, an email is sent to the system administrator.

Sample Default Output from archiver(1M) -l Command

The following shows sample output from the `archiver(1M) -l` command.

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

all path:
copy:1 arch_age:30
Archive sets:
all
copy:1 media:mo
big
copy:1 media:lt
samfs1
copy:1 media:mo
```

Log Files and Event Logging for Archive Operations

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 `sam-arfind` and `sam-arcopy` processes use the `syslog` facility and `archiver.sh` to log warnings and informational messages in a log file that contains information about each archived or automatically unarchived file.

The log file is disabled by default. Use the `logfile=` directive in the `archiver.cmd` file to enable logging and to specify the name of the log file. For more information about the log file, see About the archiver.cmd File and the `archiver.cmd`(4) man page.

The following example shows sample lines from an archiver log file. The following table defines each field in the log.

Example - Archiver Log File Lines

```
A 2001/03/23 18:42:06 mo 0004A arset0.1 9a089.1329 samfs1 118.51 162514 t0/fdn f 0 56
A 2001/03/23 18:42:10 mo 0004A arset0.1 9aac2.1 samfs1 189.53 1515016 t0/fae f 0 56
A 2001/03/23 18:42:10 mo 0004A arset0.1 9aac2.b92 samfs1 125.53 867101 t0/fai f 0 56
A 2001/03/23 19:13:09 lt SLOT22 arset0.2 798.1 samfs1 71531.14 1841087 t0/fhh f 0 51
A 2001/03/23 19:13:10 lt SLOT22 arset0.2 798.e0e samfs1 71532.12 543390 t0/fhg f 0 51
A 2003/10/23 13:30:24 dk DISK01/d8/d16/f216 arset4.1 810d8.1 qfs2 119571.301 1136048 t1/fileem f 0
0
A 2003/10/23 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.8ad qfs2 119573.295 1849474 t1/fileud f
0 0
A 2003/10/23 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.16cb qfs2 119576.301 644930 t1/fileen f
0 0
A 2003/10/23 13:30:25 dk DISK01/d8/d16/f216 arset4.1 810d8.1bb8 qfs2 119577.301 1322899 t1/fileeo
f 0 0
```

Table - Archiver Log File Fields

Field	Example Value	Content

1	A	Archive activity, as follows: • A for archived. • R for rearchived. • U for unarchived.
2	2001/03/23	Date of the archive action, in yyyy/mm/dd format.
3	18:42:06	Time of the archive activity, in hh:mm:ss format.
4	mo	Archive media type. For information about media types, see the mcf(4) man page.
5	0004A	VSN. For removable media cartridges, the volume serial name. For disk archives, the disk volume name and archive tar (1) file path.
6	arset0.1	Archive set and copy number.
7	9a089.1329	Physical position of the start of the archive file on media (tar(1) file) and file offset within the archive file, in hexadecimal format.
8	samfs1	File system name.
9	118.51	Inode number and generation number. The generation number is used in addition to the inode number for uniqueness because inode numbers are reused.
10	162514	Length of the file if the file is written on only one volume. Length of the section if the file is written on multiple volumes.
11	t0/fdn	Path and name of the file relative to the file system's mount point.
12	f	Type of file, as follows: • d for directory • f for regular file • l for symbolic link. • R for removable media file • I for segment index • S for data segment
13	0	Section of an overflowed file or segment. If the file is an overflowed file, the value is nonzero. For all other file types, the value is 0.
14	56	Equipment ordinal of the drive on which the file was archived.

Data Verification

You can enable data verification for archive copies. This feature checks for data corruption on any data that is copied to secondary and/or tertiary media.

The data verification process performs a read-after-write verification test and records a confirmation of data validity in the metadata properties for that file. The process uses the `ssum` option to mark files and directories to be verified. The normal checksum method is employed to verify copies written to tape or disk archive.

Use the `ssum -e` command to set data verification for a file or directory. Child directories inherit the data verification properties of their parent. This command forces the generation and use of checksums for archiving and staging, and prevents the release of the file until all archive copies have been created and their checksums verified. Only a superuser can set this attribute on a file or directory.

> Note
> Data verification places an additional burden on stager resources because data verification requests are placed on the stager queue in addition to normal stage requests. Data verification also leads to additional tape mounts and therefore affects archiver and stager performance. Because a file cannot be released until all archive copies are made, using data verification might also require additional disk cache.

About Recycling

Contents

About Recycling

Recycling is the process of reclaiming space on archive volumes. The recycler works with the archiver to reclaim the space occupied by unused archive copies. As users modify files, the archive copies associated with the old versions can be purged from the system. The recycler identifies the volumes with the largest proportions of expired archive copies and directs the movement of unexpired copies to different volumes. If only expired copies exist on a given volume, a site-defined action is taken. For example, a volume can be relabeled for immediate reuse or exported to offsite storage, keeping a separate historical record of file changes. Users are unaware of the recycling process.

At any time, the space on an archive volume consists of the following:

- Current data, consisting of archive images that are active
- Expired data, consisting of archive images that are no longer active
- Free space, consisting of space that is not being used by active or expired archive images

The recycler keeps the amount of space consumed by expired data to the minimum defined by site-specified parameters.

The capacity of a volume is the total amount of space for data on a volume. For example, a 10-gigabyte volume with 3 gigabytes written to it has a capacity of 10 gigabytes and 7 gigabytes of free space.
New or newly-labeled archive media starts with all its capacity as free space. As data is archived to the media, the amount of free space decreases and the amount of current data increases.

As files in the file system are changed or removed, their archive images expire and the classification of their data changes from the current data to the expired data. The physical space used by the archive images does not change. However, no file in the file system points to that space. When space is recycled, these images are removed and the space they occupied become free, available for other purposes. The goal of the recycler is to transform space used by expired data into free space without losing any current data.

The Recycle Process

The recycler and the archiver work together, as follows:

1. The recycler marks all the current archive images that are present on a volume with the `rearchive` attribute.
2. If you are archiving to removable media, the recycler marks the archive volume with the `recycle` attribute to prevent the archiver from writing any more archive images to the volume.
3. The archiver moves all the marked images to another volume. This operation is called re-archiving. After the archiver moves the current archive images from the old volume to the new volume, the old volume contains only free space and expired space. If you are archiving to removable media cartridges, you can relabel and reuse the cartridge. If you are archiving to disk, the recycler removes the file that contains the expired archive images.

The recycler is designed to run periodically, although you can run it at any time. It performs as much work as it can each time it is invoked. The recycler has to finish marking copies for re-archiving before the archiver can re-archive the files. Sometimes expired archive images, with the `rearchive` attribute set, remain on media. This can happen under the following conditions:

- The archiver does not run after the recycler marks expired archive images.
- Media is not available for the archiver to use when moving the unexpired archive images.
- Miscellaneous archiver anomalies occur.

Between executions, the recycler keeps state information in the library catalogs and the inodes. During the recycling process, you can use the `sls`(1) command and its `-D` option to display information about a file. The output from the `sls`(1) command shows whether a file is scheduled for re-archiving.

Planning for Recycling

Recycling is accomplished using two methods, depending on the type of media.

Media Types and Recycling Methods

Archive Media	Recycling Method
Removable media cartridges	By automated library

Removable media cartridges	By archive set
Disks	By archive set

For information about configuring by these methods, see Configuring the Recycler.

About Releasing

Contents

About Releasing

This section describes the releasing process and releaser operations.

Releasing is the process by which the releaser makes disk cache space available by identifying archived files and releasing their disk cache copy. This action makes room for other files to be created or staged from archive media. The releaser can release only archived files. A released file has no data on the disk cache.

The Sun Storage Archive Manager (SAM) software invokes the releaser process when a site-specified disk threshold is reached. As an alternative, you can use the `release`(1) command to release a file's disk space immediately or to set releasing parameters for a file.

You can also specify that files are released immediately after archiving, that files are never released, or that files are partially released.

Releaser Process

When file system utilization exceeds its configured high-water mark, the file system management software invokes the releaser, which does the following:

- Reads the `releaser.cmd` file and collects the directives that control the release process
- Scans the file system and collects information about each file
- Begins releasing files in priority order

A file system can contain thousands of files. Keeping track of the release priority for every file can be wasteful, because releasing only several large files might return the file system to its low-water mark. However, the releaser must examine the priority of each file or risk missing the best candidates for release. The releaser does this by identifying the first 10,000 candidates. It then discards subsequent candidates if they do not have a priority greater than the lowest-priority candidate among the first 10,000.

After the releaser has determined the priority of the first 10,000 candidates, it selects the files with the highest priority for release. After releasing each file, the releaser checks whether the file system cache utilization is below the low-water mark. If so, it stops releasing files. If not, it continues releasing the files in priority order.

If the releaser has released all 10,000 candidates and the file system is still above the low-water mark, it starts over and identifies 10,000 new candidates.

The releaser exits if it cannot find any viable candidates. This situation can occur, for example, if files do not yet have archive copies. In this case, the SAM software starts the releaser again after one minute has elapsed.

The high and low-water marks are set with the `high`=percent and `low`=percent file system mount options. For more information about these mount options, see the `mount_samfs`(1M) man page.

Releaser Concepts

This section describes concepts that are basic to the releaser process:

- Age
- Candidate
- Priority
- Weight
- Partial Release

Age

Age is the amount of time that elapsed from a given event to the present. A file's inode keeps track of the following times:

- Residence-change time
- Data-modified time
- Data-accessed time

You can view these times by using the `sls`(1) command with the `-D` option. Each time has a corresponding age. For example, if it is 10:15 a.m., a file with a modify time of 10:10 a.m. has a data-modified age of five minutes. For more information, see the `sls`(1) man page.

Candidate

A candidate is a file that is eligible to be released. A file is not a candidate under the following circumstances:

- The file is already offline.
- The file has not been archived.
- The `archiver.cmd` command file specifies the `-norelease` attribute for the file and the required copies have not yet been made.
- The file is marked as damaged.
- The file is not a directory, block, character-special file, or pipe.
- The archiver is staging the file to make an additional copy. The file becomes eligible for release after the archiver stages it.
- The age of the file is negative. This condition occurs for network file system (NFS) clients with inaccurate clock settings.
- The file is marked to never be released. You can use the `release -n` command to specify this.
- The file was staged at a time in the past that is less than the minimum residence time setting. For more information, see The min_residence_age Directive: Specifying a Minimum Residence Time.
- The file was flagged for partial release, through the `release`(1) command's `-p` option, and it is already partially released.
- The file is too small. Releasing it will not create much space.

Priority

A priority is a numeric value that indicates the rank of a candidate file based on user-supplied weights that are applied to numeric attributes of that candidate. The overall priority is the sum of two types of priority: age priority and size priority. Candidate files with numerically larger priorities are released before candidates with numerically smaller priorities.

Weight

A weight is a numeric value that biases the priority calculation to include file attributes in which you are interested and to exclude file attributes in which you are not interested. For example, if the size weight is set to 0, the size attribute of a file is excluded from the priority calculation. Weights are floating-point values from 0.0 to 1.0.

Partial Release

With partial release, a beginning portion of the file remains in disk cache while the rest of the file is released. Partial release is useful with utilities such as `filemgr`(1) that read the beginning of a file.

About Partial Releasing and Partial Staging

Releasing and staging are complementary processes. Files can be completely released from online disk cache after they are archived, or a site can specify that the beginning of a file (the stub) remain in disk cache while the remainder of the file is released. Partially releasing a file provides immediate access to data in the file stub without requiring that the file be staged. You can specify both the default partial release size and the maximum size of the stub to remain online when a file system is mounted.

You can set the through the values for partial releasing and staging using the `mount`(1M) command or in the SAM-QFS Manager. See the SAM-QFS Manager online help for more information.

The `mount`(1M) command options are as follows:

- `-o partial=n` option - Sets the default size (n) of a file stub to remain online. The `-o partial=n` setting must be less than or equal to the `-o maxpartial=n` setting.
- `-o maxpartial=n` option - Sets the maximum size (n) of a file stub to remain online.

You can specify the default stub size for a file by specifying the `-p` option on the `release`(1) command or the p option on the `sam_release`(3) library routine. To specify different-sized file stubs for different types of files or different applications, specify the `-s` option on the `release`(1) command or the s option on the `sam_release`(3) library routine. The `-s` and s values must be less than the `-o maxpartial` value used with the `mount`(1M) command when the file system was mounted.

> Note
> A partially released file takes up space on the disk equal to one DAU. For example, if the partial release file stub is set to 16K and the DAU size is 256K, the actual space consumed by the file on the disk is 256K.

Use the mount option, `-o partial_stage=n`, to establish how much of a partial release stub must be read before the rest of the file is staged. Reading past the `-o partial_stage=n` size specification initiates the stage of the file.
By default, the `-o partial_stage=n` option is set to the size of the partial release stub. Changing this value affects file staging as follows:

- If the `-o partial_stage=n` option is set to the size of the partial release stub, the default behavior prevents the file from being staged until the application reaches the end of the partial release stub. Waiting until the end of the stub is reached causes a delay in accessing the rest of the file.
- If the `-o partial_stage=n` option is set to a value smaller than the partial release stub, the file is staged after the application crosses the threshold set by the `-o partial_stage=n` option. This reduces the chance of a delay in accessing the rest of the file data.

Example - Partial Staging

In this example, a site has set the following options:

- `-o partial_stage=16` (16 kilobytes)
- `-o partial=2097152` (2 gigabytes)
- `-o maxpartial=2097152` (2 gigabytes)

The `filemgr`(1) program reads the first 8 kilobytes of a file. The file is not staged.
A video-on-demand application reads the same file. After it reads past the first 16 kilobytes of the file, the file is staged. The application continues reading while the archive tape is mounted and positioned.
When the video-on-demand application reads past two gigabytes of file data, it is reading immediately behind the staging activity. The application does not wait, because the tape mounting and positioning is done while the application reads the partial file data.

Several command-line options affect whether a file can be marked for partial release. Some options are enabled by the system administrator, and others can be enabled by individual users. The following sections describe the release characteristics that can be set by the various types of users.

- System Administrator Option Summary
- User Option Summary

Summary of System Administrator Options

As a system administrator, you can change the maximum value and default value for partial release when the file system is mounted. The mount(1M) options in the following table affect partial release. For more information about the mount(1) command, see the mount_samfs(1M) man page.

Table - Mount Options for Partial Release

Option	Effect
`-o maxpartial=n`	Determines the maximum amount of space in kilobytes that can remain in disk cache when a file is marked for partial release. The maximum value is 2,097,152 kilobytes, which is 2 gigabytes. The minimum value is 0, which disables the partial release feature is disabled so that released files are released completely, and no portion of a file remains in disk cache. Users cannot override the value specified on this option after the file system is mounted. By default, the n argument is set to 16.

-o partial=n	Sets a default amount of space in kilobytes that remains in disk cache when a user marks a file for partial release by using the release(1) command's -p option. The n argument must be at least 8, but it can be as great as the value specified for the -o maxpartial=n option. Because some applications do not need access to the entire file to complete their operations, this option ensures that applications have the beginnings of files available to them. Also, this option prevents files from being staged unnecessarily. By default, n is 16. A file that has been partially released from a disk takes up space on the disk equal to one DAU.
-o partial_stage=n	Specifies that when a partially released file is accessed, n bytes of the file must be read before the entire file is staged from the archive media. Set this value lower than the amount of the -o partial setting. For n, specify an integer from 0 to the -o maxpartial specification. By default, this is 16 or the value specified for the -o partial option.
-o stage_n_window=n	Specifies the amount of data to be staged at any one time to n. Specify an integer from 64 to 2,048,000. The default is 256 kilobytes. This option applies only to files that have the stage -n attribute set.

Summary of User Options

As a user, you can set maximum and default values for the size of a file stub that can remain in disk cache after the file is released. You can also determine whether the partial release feature is enabled for a particular file system.

By using the release(1) command and the sam_release(3) library routines, however, a user can set other release attributes and can specify the files to be marked for partial release. The command and library options that determine partial release attributes are shown in the following table. For more information, see the release(1) man page and thee sam_release(3) man page.

Table - User Release Options

Options	Effect
release(1) command and -p option or sam_release(3) library routine and p option	The -p and p options mark the named file for partial release. If these options are used, the amount of the file remaining in online disk cache after the file is released depends on the value of the -o partial=n option that was set when the file system in which the file resides was mounted. These options cannot be used to specify the number of bytes to remain online.
release(1) command and -s partial_size option or sam_release(3) library routine and s option	The -s and s options mark the named file for partial release, and they specify the amount of the file to remain in online disk cache. The arguments to the -s or s options specify the amount, in kilobytes, to remain online. A user cannot specify that the amount of a file remaining online be greater than the amount specified for the -o maxpartial=n value when the file system was mounted. If the user's value is greater than the value for the file system, the value for the file system is used, and the user's specification is ignored.

About Staging

Contents

About Staging

Staging is the process of copying file data from nearline or offline storage back to online storage.

The stager starts when the samd daemon runs. The stager has the following default behavior:

- The stager attempts to use all the drives in the library.
- The stage buffer size is determined by the media type, and the stage buffer is not locked.
- No log file is written.
- Up to 1000 stage requests can be active at any one time.
  You can customize the stager's operations for your site by inserting directives into the /etc/opt/SUNWsamfs/stager.cmd file.

When an application requires an offline file, its archive copy is staged to disk cache (if the -n option's -stage never is not set). To make the file available to an application immediately, the read operation tracks along directly behind the staging operation so that the access can begin before the entire file is staged.

Stage errors that include media errors, unavailability of media, unavailability of an
automated library, and others. If a stage error is returned, the SAM-QFS software attempts to find the next available copy of the file, if one exists and if there is a device available to read the archive copy's media.

See Configuring the Stager for details on how to configure the stager through the `stager.cmd` file.

Advanced SAM Topics

Contents

Advanced SAM-QFS Topics

This page discusses advanced topics to Sun Storage Archive Manager (SAM-QFS) system administration and usage.

Using Device Logging

The device-logging facility provides device-specific error information that you can use to analyze certain types of device problems. It can help to determine a failing sequence of events for an automated library, tape drive, or optical drive. The device-logging facility does not collect soft media errors (such as recoverable read errors).

Device-logging messages are written to individual log files. A log file is created for each automated library, for each tape and optical drive device, and for the historian. The log files are located in `/var/opt/SUNWsamfs/devlog`. The name of each log file corresponds to the name of the equipment ordinal.
For example, assume that you have a QFS file system and a single Hewlett-Packard optical library with two optical drives.
The following example shows the `mcf` file.

```
/dev/samst/c1t5u0 40 hp hp40 - /etc/opt/SUNWsamfs/hp40_cat
/dev/samst/c1t4u0 41 mo hp40 -
/dev/samst/c1t6u0 42 mo hp40 -
```

The following example shows the `/var/opt/SUNWsamfs/devlog` file. Device 43 is the historian.

```
# pwd
/var/opt/SUNWsamfs/devlog
# ls
40 41 42 43
#
```

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 great deal of device activity occurs. Initially, the device log settings are set to the following default values:

```
err retry syserr date
```

If you suspect a problem exists with one of the devices, consider enabling additional logging events for that device. Also, 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, your service provider might advise you to set the event to `all` for a device. However, in general, it is not practical to run the system with excessive logging.

The device log information is collected automatically when the `samexplorer`(1M) command is issued. This allows the file system service to review any device error information as part of problem analysis activity.

How to Enable the Device Log By Using the **`samset`**(1M) Command

- Issue the `samset`(1M) command.

  ```
  # samset devlog <eq> <event>
  ```

  For eq, specify the equipment ordinal of the device for which you want to log messages.
  For event, specify one or more of the events listed in the `samset`(1M) man page. If you specify more than one event, separate them with space characters.

How to Enable the Device Log by Editing the **`defaults.conf`** File

1. Become superuser.
2. In the `/etc/opt/SUNWsamfs/defaults.conf` file, add the `devlog` directive.

   ```
   devlog <eq> <event>
   ```

   For eq, specify the equipment ordinal of the device for which you want to log messages.
   For event, specify one or more of the events listed in the `samset`(1M) man page. If you specify more than one event, separate them with space characters.
   When a QFS file system starts, it sets the event type for each available device to `default`. You can also use the `samset`(1M) command to determine the present settings for each device log.

3. Save and close the `defaults.conf` file.
4. Use the `samd`(1M) `config` command to propagate the `defaults.conf` file changes.

   ```
   # samd config
   ```

Using Removable Media Files

You can use the `request`(1) command to manually create, write, and read files that do not use the disk cache for buffering the data. Files created in this manner are called removable media files.

> Note -
> The `request`(1) command bypasses the typical functions of the archiver.

Removable media files look like typical QFS files in that they have permissions, a user name, a group name, and size characteristics. However, their data does not reside in the disk cache. Therefore, you can create removable media files that are larger than the disk cache and write them to removable media cartridges.

The system creates an inode entry in the `.inodes` file for the file that you specify with the `request`(1) command. The QFS file systems read that information from the inode entry. Multiple removable media files can reside on the same volume.

A removable media file that spans multiple volumes is called a volume overflow file. The volume overflow feature enables a single large file to span multiple volumes on multiple cartridges. The volume overflow feature is useful if you have very large files that exceed the capacity of their chosen media.

You must read and write removable media files sequentially. The QFS file system automatically mounts the requested volume if the volume resides in an automated library defined in the `mcf` file.

The presence of a removable media file on a volume prevents that volume from being recycled. The recycler expects that only archived files reside on the particular volume that is assigned for archiving. In addition, removable media files are never archived. Removable media files are not supported over NFS.

How to Create a Removable Media or Volume Overflow File

1. Use the `tplabel`(1M) or `odlabel`(1M) command to label a tape or magneto-optical cartridge, respectively. See Labeling and Unlabeling Cartridges for details.
2. Issue the `request`(1) command. At a minimum, use the following options:

```
request -m <media-type> -v <vsn> [<vsn>/<vsn> ...] [-l <vsn-file>] <input-file>
```

Argument	Meaning
media-type	The media type of the removable media cartridge. For information about valid media-type specifications, see the `mcf` (4) man page.
vsn	The volume serial name (VSN) of the removable media cartridge. If you specify more than one VSN, you are creating a volume overflow file. You can specify up to 256 VSNs for volume overflow files. Use forward slash characters (/) to separate the vsn arguments.The VSNs specified should not be among the volumes that are used in a SAM-QFS environment for automated archiving. Archiving appends the next file to be archived to the end of the current data and moves the EOF label beyond the data.
vsn-file	An input file that contains a list of VSNs. When you have many VSNs, use an input file containing the list of VSNs
input-file	The file to be written to the removable media cartridge. This file must reside in a QFS file system.

Examples

The following command creates a removable media file:

```
# request -m lt -v aaa rem1
```

The following command creates a volume overflow file on three volumes:

```
# request -m lt -v TAPE01/TAPE02/TAPE03 large.file
```

For detailed examples of how to create removable media files, see the `request`(1) man page.

Using Segmented Files

The SAM-QFS environment supports segmented files. Segmenting files improves tape storage retrieval speed, access, and manageability for very large files. A segmented file can be larger than the physical disk cache. In this case, only part of a segmented file resides on the disk cache at any one time.

The `segment`(1) command enables you to specify the segment size. You cannot set a segment size that is smaller than the current file size.

Segmented files support tape striping. After a file is segmented, it can be striped simultaneously over multiple tape devices, which significantly reduces the time needed to store the file segments. Data access is accelerated by allowing users to retrieve only the desired file segments rather than the entire file.

Segmentation can enhance archiving efficiency because only changed portions of a file are re-archived. Segments of a file can be archived in parallel, and segmented files can be staged in parallel. This increases performance during archiving and retrieving.

Segmentation can be enabled on a file, directory, or entire file system. Segmented files support all other SAM-QFS capabilities.

> **Note -**
> The mmap function cannot take place on a segmented file. Therefore, a segmented file cannot be an executable binary.

The following sections describe how segmented files differ from nonsegmented files. For more information about segmented files, see the `segment`(1) or the `sam_segment`(3) man pages.

Archiving

For a segmented file, the archivable unit is the segment itself, not the file. All archiving properties and priorities apply to the individual segment and not to the file.

You can stripe a segment by specifying both the `- drives` and `-drivemin` parameters for the archive set in the `archiver.cmd` file. For example, assume that a 100-megabyte segmented file in the file system has segment size of 10 megabytes. If the `archiver.cmd` file defines an archive set with a `-drives` 2 directive, this file is archived to two drives in parallel. Segments 1, 3, 5, 7, and 9 are archived using the first drive, and segments 2, 4, 6, 8, and 10 are archived using the second drive.

Only segments that have been modified are archived. Up to four archive copies can be made for each segment. SAM-QFS also supports volume overflow for segments.

> **Note -**
> The index of a segmented file contains no user data. It is considered metadata and is assigned to the file system archive set.

Disaster Recovery

For information about recovering a segmented file in the event of a disaster, see SAM-QFS Troubleshooting.

Using System Error Facility Reporting

The system error facility (SEF) reporting system captures log sense data from tape devices in an automated library, writes it to a log file, and translates it into human-readable form. This utility consists of the following:

- A log file containing data from tape device log sense pages.
- A command, `sefreport`(1M), for writing the log file to `stdout` in a human-readable format. This log file can be used as input to a user-supplied analysis script.
  The log sense pages are different from vendor to vendor. For the meanings of the parameter codes, control bits, and parameter values, see the vendor documentation for each specific device.

SEF is not supported for stand-alone tape drives. SEF reporting is most useful for older SCSI-2 devices that do not support the `tapealert`(1M) functionality. For more information, see the `tapealert`(1M) man page.

How to Enable SEF Reporting

1. Become superuser.
2. Issue the `mkdir`(1) command to create the SEF directory. For example:

```
# mkdir /var/opt/SUNWsamfs/sef
```

3. Use the `touch`(1) command to create the log file.
   You can enable SEF reporting any time after installation by creating the `sefdata` log file. Initially, the SEF log file must be empty. The following command shows the SEF log file being created in the default location.

```
# touch /var/opt/SUNWsamfs/sef/sefdata
```

4. Use the `samd stop` and `samd start` commands to initialize SEF reporting.

```
samcmd aridle
samcmd stidle
```

Wait until the tape drives are idle. Then unload the tape drives:

For eq, specify the Equipment Number of the drive.

1. Use the samd(1M) unload command to unload all removable media:

```
samcmd unload <eq>
```

Repeat the above commands for each tape drive.

```
# samd stop
# samc config
# samd start
```

SEF data is appended to the log file as it is generated.

> Note -
> SEF reporting is enabled as long as the sefdata log file exists. To disable SEF reporting, rename or remove this file.

You can configure SEF reporting to log and read log sense data from an alternate location. For more information about reading log sense data from an alternate location, see the sefreport(1M) man page.

How to Generate SEF Report Output

The SEF report output consists of header lines and log sense data.

1. Verify that /opt/SUNWsamfs/sbin is in your command path.
2. Use the sefreport(1M) command to generate SEF output.
   The following are the most commonly used options with the sefreport(1M) command:

- The -d option. The -d option generates additional device information. It writes an additional header line that contains the equipment ordinal and path name to the device for each record. This process makes searching for and locating SEF records that pertain to a specific device easier.
- The -v option or the -t option. Do not specify the -t and -v options on the same command line. They are mutually exclusive.
  - The -v option generates information in verbose mode. It appends information regarding the equipment ordinal, page code, and VSN to each line of a record. This method enables selecting only those lines that pertain to a specific device or a specific volume.
  - The -t option generates log sense output with text descriptions. For each line of log sense data output, the report includes an additional string containing the equipment ordinal, page code, VSN, and parameter code description.
    For example, the following SEF command reads the SEF log file from the default location, writes the device number and path name for each device, and generates output:

```
# sefreport -d /var/opt/SUNWsamfs/sef/sefdata > sef.output
```

Example: Content of sef.output File

```
Record no. 1
Mon Mar 26 11:17:48 2001 STK 9840 1.25 VSN 002981
Eq no. 32 Dev name /dev/rmt/1cbn
PAGE CODE 2
param code control param value
00h 74h 0x0
```

```
01h 74h 0x0
02h 74h 0x0
03h 74h 0x0
04h 74h 0x0
05h 74h 0x40050
06h 74h 0x0
PAGE CODE 3
param code control param value
00h 74h 0x0
01h 74h 0x0
02h 74h 0x0
03h 74h 0x0
04h 74h 0x0
05h 74h 0x140
06h 74h 0x0
PAGE CODE 6
param code control param value
00h 74h 0x0
Record no. 2
Mon Mar 26 11:30:06 2001 STK 9840 1.25 VSN 002999
Eq no. 31 Dev name /dev/rmt/0cbn
PAGE CODE 2
param code control param value
00h 74h 0x0
01h 74h 0x0
02h 74h 0x0
03h 74h 0x0
04h 74h 0x0
05h 74h 0x1400a0
06h 74h 0x0

PAGE CODE 3
param code control param value
00h 74h 0x0
01h 74h 0x0
02h 74h 0x0

03h 74h 0x0
04h 74h 0x0
05h 74h 0x190
06h 74h 0x0
PAGE CODE 6
param code control param value
00h 74h 0x0
Record no. 3
Mon Mar 26 11:30:23 2001 STK 9840 1.25 VSN 002981
Eq no. 32 Dev name /dev/rmt/1cbn
PAGE CODE 2
param code control param value
00h 74h 0x0
01h 74h 0x0
02h 74h 0x0
03h 74h 0x0
04h 74h 0x0
05h 74h 0x18400f0
06h 74h 0x0

PAGE CODE 3
param code control param value
00h 74h 0x0
01h 74h 0x0
02h 74h 0x0
03h 74h 0x0
04h 74h 0x0
05h 74h 0x1e0
06h 74h 0x0
PAGE CODE 6
param code control param value
00h 74h 0x0
.
```

```
.
.
```

For more information about the SEF log file, including its content and format, see the `sefdata`(4) man page. For more information about optional SEF report formats, see the `sefreport`(1M) man page.

Managing the SEF Log File

You manage the SEF log file just as you manage any other SAM-QFS log file. You can run a `cron`(1) job periodically to save the current log file to another location, to delete old SEF files, to create new (empty) SEF files, or to perform other file management tasks.
You can also use the `log_rotate.sh`(1M) utility to rotate this log file.
For more information about tools for managing the SEF log file, see the `cron`(1) or `log_rotate.sh`(1M) man pages.

SEF **sysevent** Functionality

In addition to using the SEF log file, you can use the Solaris `sysevent` feature to obtain tape drive SCSI log sense error counter pages 2 and 3 for media analysis. By default, the SEF `sysevent` feature is enabled and set to poll once before unload. The SEF `sysevent` behavior is controlled by `defaults.conf` and `samset`.

In the `defaults.conf` file, the `sef` parameter can be used to enable SEF `sysevent` feature by equipment ordinal, or to specify the log sense polling frequency. For more information, see the `defaults.conf`(4) manu page.

How to Create the SEF **sysevent** Handler

1. Create a `/var/tmp/xx` file similar to the following.

```
#!/bin/ksh
echo "$@" >> /var/tmp/xx.dat
exit 0
```

2. Make the `/var/tmp/xx` file executable.

```
# chmod a+rwx /var/tmp/xx
```

Add the SEF `sysevent` handler to the `syseventd` file by adding the following information.

```
# syseventadm add -vSUNW -pSUNWsamfs -cDevice -sSEF
/var/tmp/xx \"\$VENDOR\" \"\$PRODUCT\" \"\$USN\" \"\$REV\" \
$TOD \$EQ_ORD \"\$NAME\" \$INQ_TYPE \"\$MEDIA_TYPE\" \"\$VSN\" \
\$LABEL_TIME \$LP2_PC0 \$LP2_PC1 \$LP2_PC2 \$LP2_PC3 \$LP2_PC4 \
\$LP2_PC5 \$LP2_PC6 \$LP3_PC0 \$LP3_PC1 \$LP3_PC2 \$LP3_PC3 \
\$LP3_PC4 \$LP3_PC5 \$LP3_PC6 \$WHERE \$sequence
# syseventadm restart
```

This command creates the `/etc/sysevent/config/SUNW,SUNWsamfs,Device,sysevent.conf` file containing the SEF `sysevent` handler `/var/tmp/xx` and loads the event handler into the `syseventd` daemon.

3. To load the SEF sysevent handler, issue the following command:

```
pkill -HUP syseventd
```

For more information, see the `sefsysevent`(4) man page.

Archive Directives (archiver.cmd)

Contents

Archive Directives (**`archiver.cmd`**)

This section provides details about the archive directives.

Global Archiving Directives

Global directives control the overall archiver operation and enable you to optimize operations for your site. You can add global directives directly to the `archiver.cmd` file, or you can specify them using the SAM-QFS Manager software. For more information on using SAM-QFS Manager to set global directives, see the SAM-QFS Manager online help.

Specify the global directives before you specify any file system directives (`fs=` directives). The archiver issues a message if it detects a global directive located after an `fs=` directive.

You can identify a global directive in the `archiver.cmd` file by either the equal sign (=) in the second field or the absence of additional fields. The following global directives are supported:

- `archivemeta` Directive
- `archmax` Directive
- `bufsize` Directive
- `drives` Directive
- `examine` Directive
- `interval` Directive
- `logfile` Directive
- `notify` Directive
- `ovflmin` Directive
- `scanlist_squash` Directive
- `setarchdone` Directive
- `wait` Directive

`archivemeta` Directive: Controlling Whether Metadata Is Archived

The `archivemeta` directive controls whether file system metadata is archived. If files are often moved around and there are frequent changes to the directory structures in a file system, archive the file system metadata. In contrast, if the directory structures are very stable, you can disable metadata archiving and reduce the actions performed by removable media drives. By default, metadata is not archived.

This directive has the following format:

```
archivemeta=<state>
```

For state, specify either on or `off`. The default is `off`.

The archiving process for metadata depends on whether you are using a Version 1 or a Version 2 superblock, as follows:

- For Version 1 file systems, the archiver archives directories, removable media files, segment index inodes, and symbolic links as metadata.
- For Version 2 file systems, the archiver archives directories and segment index inodes as metadata. Removable media files and symbolic links are stored in inodes rather than in data blocks. They are not archived. Symbolic links are archived as data.

`archmax` Directive: Controlling the Size of Archive Files

The `archmax` directive specifies the maximum size of an archive file. User files are combined to form the archive file. After the target-size value is met, no more user files are added to the archive file. Large user files are written in a single archive file.
To change the defaults, use the following directive:

```
archmax=<media> <target-size>
```

Argument	Meaning
media	The media type. For the list of valid media types, see the `mcf`(4) man page.
target-size	The maximum size of the archive file. This value is media-dependent. By default, archive files written to optical disks are no larger than 5 megabytes. The default maximum archive file size for tapes is 512 megabytes.

Setting large or small sizes for archive files has advantages and disadvantages. For example, if you are archiving to tape and `archmax` is set to a large size, the tape drive stops and starts less often. However, when writing large archive files, a premature end-of-tape causes a large amount of tape to be wasted. As a best practice, do not set the `archmax` directive to be more than 5 percent of the media capacity.

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

> Note -
> The `archmax` directive is not a valid directive for archive sets that are archived to the Sun StorageTek 5800 media type.

`bufsize` Directive: Setting the Archiver Buffer Size

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

```
bufsize=<media> <buffer-size> [lock]
```

Argument	Meaning
media	The media type. For the list of valid media types, see the `mcf`(4) man page.
buffer-size	A number from 2 through 1024. The default is 4. This value is multiplied by the dev *`blksize`* value for the media type, and the resulting buffer size is used. The _dev _`blksize` value is specified in the `defaults.conf` file. For more information about this file, see the `defaults.conf`(4) man page.

lock	Indicates whether the archiver can use locked buffers when making archive copies. If lock is specified, the archiver sets file locks on the archive buffer in memory for the duration of the sam-arcopy(1M) operation. This action avoids the overhead associated with locking and unlocking the buffer for each I/O request and results in a reduction in system CPU time. The lock argument must be specified only on large systems with large amounts of memory. Insufficient memory can cause an out-of-memory condition. The lock argument is effective only if direct I/O is enabled for the file being archived. By default, lock is not specified and the file system sets the locks on all direct I/O buffers, including those for archiving. For more information about enabling direct I/O, see the setfa(1) man page, the sam_setfa(3) library routine man page, or the -O forcedirectio option on the mount_samfs(1M) man page.

You can specify a buffer size and a lock for each archive set basis by using the archive set copy parameters, -bufsize and -lock. For more information, see Archive Set Copy Parameters.

drives Directive: Controlling the Number of Drives Used for Archiving

By default, the archiver uses all of the drives in an automated library for archiving. To limit the number of drives used, use the drives directive. This directive has the following format:

```
drives=<auto-lib> <count>
```

Argument	Meaning
auto-lib	The family set name of the automated library as defined in the mcf file.
count	The number of drives to be used for archiving activities.

Also see the archive set copy parameters, -drivemax, -drivemin, and -drives described in Specifying the Number of Drives for an Archive Request: -drivemax, -drivemin, and -drives.

examine Directive: Controlling Archive Scans

New files and files that have changed are candidates for archiving. The archiver finds such files through one of the following methods:

- Continuous archiving, in which the archiver works with the file system to detect file changes immediately
- Scan archiving, in which the archiver scans the file system periodically for files that need to have changed
  This directive has the following format:

```
examine=<method>
```

method Value	Meaning
noscan	Specifies continuous archiving. After the initial scan, directories are scanned only when the content changes and archiving is required. Directory and inode information is not scanned. This archiving method provides better performance than scan archiving, particularly for file systems with more than 1,000,000 files. Default.
scan	Specifies scan archiving. The initial file system scan is a directory scan. Subsequent scans are inode scans.
scandirs	Specifies scan archiving on directories only. If the archiver finds a directory with the no_archive attribute set, the directory is not scanned. If you have files that do not change, place them in this type of directory to reduce the amount of time spent on archiving scans.
scaninodes	Specifies scan archiving on inodes only.

interval Directive: Specifying an Archive Interval

The archiver runs periodically to examine the status of all mounted archived-enabled file systems. The timing is controlled by the archive interval, which is the time between scan operations on each file system. To change the time, use the interval directive.

The interval directive initiates full scans only when continuous archiving is not set and no startage, startsize, or startcount parameters have been specified. If continuous archiving is set (examine=noscan), the interval directive acts as the default startage value. This directive has the following format:

```
interval=<time>
```

For time, specify the amount of time you want between scan operations on a file system. By default, time is interpreted in seconds and has a value of 600, which is 10 minutes. You can specify a different unit of time, such as minutes or hours.

If the archiver receives the `samu`(1M) utility's `:arrun` command, it begins scanning all file systems immediately. If the `examine=scan` directive is also specified in the `archiver.cmd` file, a scan is performed after `:arrun` or `:arscan` is issued.

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

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

`logfile` Directive: Specifying An Archiver Log File

The archiver can produce a log file that contains information about each file that is archived, re-archived, or automatically unarchived. The log file is a continuous record of archival action. By default, this file is not produced. To specify a log file, use the `logfile` directive. This directive has the following format:

```
logfile=<pathname>
```

For pathname, specify the absolute path and name of the log file. The `logfile` directive can also be set for an individual file system.

Example - Backing Up an Archiver Log File

Assume that you want to back up the archiver log file every day by copying the previous day's log file to an alternate location. Be sure to perform the copy operation when the archiver log file is closed, not while it is open for a write operation.

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

`notify` Directive: Renaming the Event Notification Script

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

```
notify=<filename>
```

For filename, specify the name of the file containing the archiver event notification script or the full path to this file. The default file name is `/etc/opt/SUNWsamfs/scripts/archiver.sh`

The archiver executes this script to process various events in a site-specific manner. The script is called with one of the following keywords for the first argument: `emerg`, `alert`, `crit`, `err`, `warning`, `notice`, `info`, and `debug`.

Additional arguments are described in the default script. For more information, see the `archiver.sh`(1M) man page.

`ovflmin` Directive: Controlling Volume Overflow

When volume overflow is enabled, the archiver can create archived files that span multiple volumes. When a file size exceeds the specified minimum size, the archiver writes the remaining portion of this file to another volume of the same type. The portion of the file written to each volume is called a section.
The `sls`(1) command lists the archive copy, showing each section of the file on each volume.

> Note -
> Use volume overflow with caution after assessing its effect on your site. Disaster recovery and recycling are much more difficult with files that span volumes. For more information, see SAM-QFS Troubleshooting and the `request`(1) man page.

The archiver controls volume overflow through the `ovflmin` directive. By default, volume overflow is disabled. To enable volume overflow, use the `ovflmin` directive in the `archiver.cmd` file. This directive has the following format:

```
ovflmin = <media> <minimum-file-size>
```

Argument	Meaning
media	The media type. For a list of valid media types, see the `mcf`(4) man page.
minimum-file-size	The minimum file size that you want to trigger the volume overflow.

The `ovflmin` directive can also be set for an individual archive set.
Volume overflow files do not generate checksums. For more information on using checksums, see the `ssum`(1) man page.

Examples of Volume Overflow

A site has many files with a significant mo media cartridge fraction length (such as 25 percent). These files leave unused space on each volume. To use volume space efficiently, set `ovflmin` for mo media to a size slightly smaller than the size of the smallest file. The following directive sets it to 150 megabytes:

```
ovflmin=mo 150m
```

In this example, two volumes are loaded for archiving and staging the files because each file overflows onto another volume.

The following example shows the archiver log file when volume overflow is enabled. The file `file50` spans three volumes with VSNs of `DLT000`, `DLT001`, and `DLT005`. The position on the volume and the size of each section is indicated in the seventh and tenth fields respectively (7eed4.1 and 477609472 for the first volume).
For a complete description of the archiver log entry, see the `archiver`(1M) man page.

```
A 97/01/13 16:03:29 lt DLT000 big.1 7eed4.1 samfs1 13.7 477609472 00 big/file50 0 0
A 97/01/13 16:03:29 lt DLT001 big.1 7fb80.0 samfs1 13.7 516407296 01 big/file50 0 1
A 97/01/13 16:03:29 lt DLT005 big.1 7eb05.0 samfs1 13.7 505983404 02 big/file50 0 2
```

This portion of the archiver log file matches the `sls -D` output for file `file50`, as shown in the following example.

```
# sls -D file50
file50:
mode: -rw-rw---- links: 1 owner: gmm group: sam
length: 1500000172 admin id: 7 inode: 1407.5
offline; archdone; stage -n
copy1: ---- Jan 13 15:55 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
```

`scanlist_squash` Directive: Controlling Scanlist Consolidation

The `scanlist_squash` parameter controls scanlist consolidation. The default setting is `off`. This parameter can be either global or file-system-specific.
When this option is enabled, the scan list entries for files in two or more subdirectories with the same parent directory that need to be scanned by `sam-arfind` at a much later time are consolidated. These directories are combined upwards to the common parent, which results in a deep recursive scan of many subdirectories. This consolidation can cause a severe performance penalty if archiving on a file system that has a large number of changes to many subdirectories.

setarchdone Directive: Controlling the Setting of the **archdone** Flag

The setarchdone parameter is a global directive that controls the setting of the archdone flag when the file is examined by sam-arfind. This directive has the following format:

```
setarchdone=on|off
```

When all archive copies for a file have been made, the archdone flag is set for that file to indicate that no further archive action is required. During directory scans, the archdone flag is also set for files that will never be archived. Because evaluating whether a file will ever be archived can affect performance, the setarchdone directive gives you control over this activity. This directive controls the setting of the archdone flag only on files that will never be archived. It does not affect the setting of the archdone flag after archive copies are made.
The default setting for the directive is off if the examine directive is set to scandirs or noscan.

wait Directive: Delaying Archiver Startup

The wait directive causes the archiver to wait for a start signal from samu(1M) or SAM-QFS Manager. By default, the archiver begins archiving when started by sam-fsd(1M). This directive has the following format:

```
wait
```

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

File System Directives

Archiving controls apply to all file systems. However, you can confine some controls to an individual file system. When the archiver encounters an fs= directive in the archiver.cmd file, all subsequent directives are applied to that specific file system. Therefore, in the archiver.cmd file, place any fs= directives after the general directives.

You can specify fs= directives either by editing the archiver.cmd file as described in the following sections, or by using the SAM-QFS Manager software. See the SAM-QFS Manager online help for more information.

fs Directive: Specifying the File System

Use the fs= directive to specify actions for a particular file system.
For instance, you can use this directive to create a different log file for each file system. This directive has the following format:

```
fs=<fsname>
```

For fsname, specify the file system name as defined in the mcf file.

The general directives and archive set association directives that occur after a fs= directive apply only to the specified file system until another fs= directive is encountered.

Global Directives As File System Directives

Several directives can be specified both as global directives for all file systems and as directives specific to only one file system:

- interval directive
- logfile directive
- scanlist_squash directive
- wait directive

Archive Copy Directives

By default, the archiver writes a single archive copy for files in the archive set when the archive age of the file is four minutes. To change the default behavior, use archive copy directives. Archive copy directives must appear immediately after the archive set assignment directive to which they pertain.

The archive copy directives begin with a copy-number value of 1, 2, 3, or 4. The digit is followed by one or more arguments that specify archive characteristics for that copy. Each archive copy directive has the following format:

```
<copy-number> [ -release | -norelease ] [<archive-age>] [<unarchive-age>]
```

You can specify archive copy directives either by editing the `archiver.cmd` file as described in the following sections, or by using the SAM-QFS Manager software. For more information, see the SAM-QFS Manager online help.

The following sections describe the archive copy directive arguments.

`-release` Directive: Releasing Disk Space After Archiving

To specify that the disk space for files is released after an archive copy is made, use the `-release` directive after the copy number. This directive has the following format:

```
-release
```

Example - **`archiver.cmd`** File Using the **`-release`** Directive

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

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

`-norelease` Directive: Delaying Disk Space Release

The `-norelease` option prevents the automatic release of disk cache until all copies marked with `-norelease` are made. The `-norelease` directive makes the archive set eligible to be released after all copies have been archived, but the files are not released until the releaser is invoked and selects them as release candidates. This directive has the following format:

```
-norelease
```

Using the `-norelease` directive on a single copy has no effect on automatic releasing.

Example - **`archiver.cmd`** File Using the **`-norelease`** Directive

The following example specifies an archive set named `vault_tapes`. Two copies are created and then the disk cache associated with this archive set is released.

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

Using `-release` and `-norelease` Together

To make sure that disk space is released immediately after all copies of an archive set have been archived, use the `-release` and `-norelease` directives together. The combination of `-release` and `-norelease` causes the archiver to release the disk space immediately after all copies having this combination are made, rather than waiting for the releaser to be invoked.

Setting the Archive Age

Change the timing for archiving files by specifying the archive age. Specify the time with a suffix character such as `h` for hours or `m` for minutes as shown in Table - File Age Suffixes.

Example - **archiver.cmd** File That Specifies the Archive Age

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

```
ex_set data
1 1h
```

Unarchiving Automatically

If you specify more than one archive copy of a file, you can unarchive all but one of the copies automatically. You can do this when the files are archived to various media using various archive ages.

Example - **archiver.cmd** File that Specifies the Unarchive Age

The following example shows the directive that specifies the unarchive age. The first copy of the files in the path `home/users` is archived six minutes after modification. When the files are 10 weeks old, the archiver creates the second and third archive copies and unarchives the first copy.

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

For more ways to control unarchiving, see Controlling Unarchiving.

Specifying More Than One Copy for Metadata

If more than one copy of metadata is required, place copy definitions in the `archiver.cmd` file immediately after the `fs=` directive.

Example - **archiver.cmd** File that Specifies Multiple Metadata Copies

In this example, one copy of the metadata for the `samfs7` file system is made after 4 hours and a second copy is made after 12 hours.

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

File system metadata includes path names in the file system. If you have frequent changes to directories, the new path names cause the creation of new archive copies and results in frequent loading of the volumes specified for metadata.

Archive Set Directives (archiver.cmd)

Contents

Archive Set Directives (archiver.cmd)

This section provides information about the archive set directives:

About Archive Set Directives

The following archive sets are available by default:

- Reserved archive set: `no_archive` and `allsets`.
    - The `no_archive` archive set is defined by default. Files assigned to this archive set are never archived. For example, files in a temporary directory can be assigned to the `no_archive` archive set.
    - The `allsets` archive set defines parameters that apply to all archive sets.
- Each file system has a default archive set with the same name that cannot be changed. These archive sets are reserved for control structure information. For each file system, both the metadata and data files are archived. The file system archive set includes the directory and link information and any files that are not included in another archive set.

By default, files are archived as part of the archive set named for the file system. However, you can create archive sets for files that share common criteria such as size, ownership, group, or directory location. If a file does not match one of the specified archive sets, it is archived as part of the default archive set. A file in the file system can be a member of only one archive set. All files in an archive set are copied to the volumes associated with that archive set.

Archive files are compatible with the standard UNIX `tar`(1) format for data compatibility with the Solaris OS and other UNIX systems. If a complete loss of your SAM-QFS environment occurs, the `tar`(1) format allows file recovery using standard UNIX tools and commands.

The characteristics of archive set are controlled by the archive set directives. These directives are arranged in the following categories:

- Assignment directive define archive sets: the destination of the archive copy, how long the copy is kept archived, and how long the software waits before archiving the data
- Copy parameters define how each archive set is archived: The archiving process copies the data necessary for file system operations, including directories, symbolic links, the index of segmented files, and archive media information.
- VSN association directives assign volumes to archive sets.
- VSN pools directives define a collection of volumes.

You can create archive sets either by editing the `archiver.cmd` file as described in the following sections, or by using the SAM-QFS Manager software. In the SAM-QFS Manager, an archive policy defines an archive set. For more information, see the SAM-QFS Manager online help.

Archive Set Assignment Directive

The archive set assignment directive select files according to shared characteristics to include in archive sets. Each archive set assignment directive has the following format:

```
<archive-set-name> <path> [<search-criterion1> <search-criterion2> ... ] [<file-attribute1>
<file-attribute2> ...]
```

Argument	Meaning

archive-set-name	A site-defined name for the archive set. A best practice is to define a name that identifies the common characteristics of the files belonging to the archive set. The name has the following requirements: • Maximum of 29 characters • Uppercase and lowercase alphabetic characters, numbers 0-9, and the underscore character (_). • No other special characters or spaces are allowed. • The first character must be a letter. • You cannot create an archive set with the name of one of the reserved archive sets, `no_archive` or `allsets`. To prevent archiving of the files in an archive set, specify_ the name as *`no_archive`*
path	Specify the path relative to the mount point of the file system so that files in the directory specified by path and its subdirectories are included in this archive set. To include all of the files in a file system, use a period (.). A leading slash (/) is not allowed in the path.
search-criterion1 search-criterion2	Zero, one, or more search criteria can be specified to restrict the archive set to files that meet the criteria. Search criteria include file age, file size, file ownership, and file name.
file-attribute1 file-attribute2	Zero, one, or more file attributes can be specified. These file attributes are set for files as the `sam-arfind` process scans a file system during archiving.

Examples - Archive set assignment directives.

In this example, all files in the user account hmk are archived in a separate archive set. All files over 1 MB in size in the directories in the `xray` group are archived in the `datafiles` archive set. All other files are archived in the `system` archive set.

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

The following example shows directives that prevent archiving of files in a `tmp` directory at any level and regardless of the directory in which the `tmp` directory resides within the file system.

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

File Age search-criterion: **`-access`** and **`-nftv`**

To use the last time a file was opened to define assignment to an archive set, use the `-access` age characteristic as one of the search-criterion arguments.
This characteristic causes files that have not been accessed within the value of age to be re-archived to different, less-expensive media. For age, specify an integer followed by one of the suffixes shown in the following table.

Table - File Age Suffixes

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

When determining age, the software validates the access and modification times for files to ensure that these times are greater than or equal to the file creation time, and less than or equal to the time at which the file is examined. For files that have been migrated into a directory, this validation might not result in the desired behavior. Use the `-nftv` (no file time validation) parameter in these situations to prevent the validation of file access and modification times.

File Age search-criterion: **-after**

Use the `-after` date-time characteristic to include files that have been modified or created recently into the same archive set. Only files created or modified after the date indicated are included in the archive set. Specify the date and time in the following format:

```
<YYYY>-<MM>-<DD>[T<hh>:<mm>:<ss>][Z]
```

If the time is not specified, the default time is 00:00:00. If the `Z` is included, the time is Coordinated Universal Time (UTC). If the `Z` is not included, the time is local time.

File Size search-criterion: **-minsize** and **-maxsize**

Use the `-minsize` size and `-maxsize` size characteristics to restrict membership in an archive set to those over or under a specified size. For size, specify an integer followed by one of the letters shown in the following table.

Table - **-minsize** and **-maxsize** size Suffixes

Letter	Meaning
b	Bytes
k	Kilobytes
M	Megabytes
G	Gigabytes
T	Terabytes
P	Petabytes
E	Exabytes

Example - Using the **-minsize** and **-maxsize** Characteristics

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

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

Owner and Group search-criterion: **-user** and **-group**

To restrict membership in an archive group to ownership and group affiliation, use the `-user` name and `-group` name characteristics.

Example - Using the **-user** and **-group** Directive

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

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

File Name search-criterion Using Pattern Matching: **-name** regex

To specify that file names are used for assignment to an archive set, use `-name` regex characteristic, which specifies that any complete path matching the regular expression regex is to be a member of the archive set.

All files beneath the selected directory (with their specified paths relative to the mount point of the file system) go through pattern matching.

Therefore, you can specify patterns in the -name regex field to match both file names and path names.

The regex argument follows the conventions outlined in the regexp(5) man page. Regular expressions do not follow the same conventions as UNIX wildcards.

Examples - **-name**

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

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

The following directive selects files that start with the characters GEO for the satellite archive set:

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

The following directive prevents any file ending with .o from being archived:

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

Example 1 - Pattern Matching with Regular Expression

The archive set assignment directive in the following example does not archive fred.* in the user directories or subdirectories.

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

As a result, the following files are not archived:

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

The following files are archived:

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

Example 2 - Pattern Matching with Regular Expression

The archive set assignment directive in the following example does not archive fred.* in the user home directories but it does archive fred.* in the user subdirectories and in the directory share/marketing. In this case, a user home directory is anything from share/marketing/ until the next slash character "/".

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

The following files are not archived:

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

The following files are archived:

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

Release and Stage file-attributes: -release and -stage

You can set the release and stage attributes associated with files within an archive set by using the `-release` and `-stage` options, respectively. Both of these settings override any existing stage or release attributes.

See How to Specify Release Attributes for All Files in an Archive Set for information about the `-release` attribute.

The `-stage` option has the following format:

```
-stage <attribute>
```

Attribute	Meaning
a	Stage the files in this archive set associatively.
d	Reset to default.
n	Never stage the files in this archive set.

Example - Archive Sets and File Attributes

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

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

Membership Conflicts in Archive Sets

When the selection of a file for inclusion in an archive set is ambiguous, the archiver uses the following rules:

- The membership definition occurring first in the archive set is chosen. Place the most restrictive assignment directives early in the `archiver.cmd` file.
- Membership definitions local to a file system are chosen before any global definitions.
- A membership definition that exactly duplicates a previous definition is noted as an error.

The archiver evaluates the file-system-specific directives before evaluating the global directives. Therefore, files can be assigned to a local archive set (including the no_archive archive set) instead of being assigned to a global archive. This result has implications for global archive set assignments such as no_archive.

Example - **archiver.cmd** File With Membership Conflicts

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

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

Example - **archiver.cmd** File Without Membership Conflicts

The following example shows the directives to use to ensure that no .o files are archived in the two file systems.

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

Archive Set Copy Parameters

The archive set copy parameters define how each archive set is archived: the data files, directories, symbolic links, the index of segmented files, and archive media information.
The archive set copy parameters section of the archiver.cmd file begins with the params directive and ends with the endparams directive. The following example shows the format for copy parameters for an archive set.

```
params
<archive-set-name>.<copy-number>[R] [<-param1> <-param2> ...]
.
.
.
endparams
```

Table - Arguments for the Archive Set Copy Parameters

Argument	Meaning
archive-set-name	A site-defined name for the archive set. A best practice is to define a name that identifies the common characteristics of the files belonging to the archive set. The name has the following requirements: • Maximum of 29 characters • Uppercase and lowercase alphabetic characters, numbers 0-9, and the underscore character (_). • No other special characters or spaces are allowed. • The first character must be a letter.
.	A period (.) character. Used to separate archive-set-name from copy-number.
copy-number	An integer that defines the archive copy number: 1, 2, 3, or 4.
R	Specifies that the parameters being defined are for re-archived copies of this archive set. For example, you can use the R and specify VSNs in the -param1 argument to direct re-archived copies to specific volumes.

-param1 -param2	One or more parameters such as maximum size, buffer size, number of drives, and son on. The following subsections describe the parameters than can be specified between the `params` and `endparams` directives.

To set default directives for all archive sets, specify directives for the archive set `allsets` archive set. The `allsets` directives must precede the directives for archive set copies because parameters set for individual archive set copies override parameters set for the `allsets` directive. For more information about the `allsets` archive set, see the `archiver.cmd`(4) man page.

You can specify archive set copy parameters by editing the `archiver.cmd` file as described in the following sections or by using the SAM-QFS Manager software. For more information, see the SAM-QFS Manager online help.

The following subsections describe all archive set processing parameters, with the exception of disk archiving parameters. For information about disk archiving parameters, see About Disk Archiving.

Controlling the Size of Archive Files: -archmax

The `-archmax` parameter sets the maximum file size for an archive set. This parameter has the following format:

```
-archmax <target-size>
```

This parameter is very similar to the `archmax` global directive. For information about that directive and the values to enter for target-size, see archmax Directive: Controlling the Size of Archive Files.

Setting the Archiver Buffer Size: -bufsize

By default, a file being archived is stored in memory in a buffer of a default size for the media type before being written to archive media. Use the `-bufsize` directive to specify a buffer size. A custom size can improve performance. This parameter has the following format:

```
-bufsize=<buffer-size>
```

The default buffer size is 4, indicating that the actual buffer size is 4 multiplied by the `devblksize` value for the media type. Specify a number from 2 to 32. The `devblksize` value is specified in the `defaults.conf` file.
For more information about this file, see the `defaults.conf`(4) man page.

Example - Buffer Size: -bufsize

```
myset.1 -bufsize=6
```

This parameter is similar to the `bufsize=`media buffer-size global directive. For more information about that directive, see bufsize Directive: Setting the Archiver Buffer Size.

Specifying the Number of Drives for an Archive Request: -drivemax, -drivemin, and -drives

By default, the archiver uses one media drive to archive the files of one archive set. When an archive set has many files or large files, using more than one drive is advantageous. In addition, if the drives in your automated library operate at different speeds, use of multiple drives can balance these variations and increase archiving efficiency. The drive directives have the following formats:

```
-drivemax <max-size>
-drivemin <min-size>
-drives <number>
```

Argument	Meaning
max-size	The maximum amount of data to be archived using one drive.

min-size	The minimum amount of data to be archived using one drive. The default is the `-archmax` target-size value (if specified) or the default value for the media type. If you specify the `-drivemin` min-size directive, the SAM-QFS software uses multiple drives only when enough activity occurs to warrant it. As a guideline, set min-size to be large enough to cause the transfer time to be significantly longer than the cartridge change time (load, position, unload).
number	The number of drives to be used for archiving this archive set. The default is 1.

An archive request is evaluated against the parameters that are specified, as follows:

- If an archive request is less than the value of min-size, only one drive is used to write an archive request.
- If an archive request is larger than the value of min-size, the archive request is evaluated against min-size and the appropriate number of drives is scheduled up to the full number of drives specified.
- If the value of min-size is 0, an attempt is made to split the archive request among the full number of drives specified.

When you use the `-drives` parameter, multiple drives are used only if data that is more than the value of min-size is to be archived. The number of drives to be used in parallel is the lesser of the following two values:

- The size of the archive request divided by the value of min-size
- The number of drives specified by the `-drives` parameter

Use the `-drivemin` and `-drives` parameters when you want to divide an archive request among drives but do not want to have all the drives busy with small archive requests. This situation can occur with very large files.

To set these parameters, consider file creation rates, the number of drives, the time it takes to load and unload drives, and drive transfer rates. For example, a site splits an archive set named `bigfiles` across five drives. This archive set could be split as shown in the following table.

Archive Set Size	Number of Drives
< 20 gigabytes	1
> 20 gigabytes to < 30 gigabytes	2
> 30 gigabytes to < 40 gigabytes	3
> 40 gigabytes to < 50 gigabytes	4
> 50 gigabytes	5

Example - Directives Used to Split an Archive Request Over Multiple Drives

The following example how to split the archive requests of 10 GB or more over five drives.

```
params
bigfiles.1 -drives 5 -drivemin 10G
endparams
```

In addition, the following line ensures that two drives are used to archive the files when the total size of the files in archive set `huge_files.2` is equal to or greater than two times `drivemin` for the media.

```
huge_files.2 -drives 2
```

Maximizing Space on a Volume: -fillvsns

By default, the archiver selects a volume with enough space for all files when it writes an archive copy. This action results in volumes not being filled to capacity. When `-fillvsns` is specified, the archiver separates the archive request into smaller groups and can use different volumes.

Specifying Archive Buffer Locks: -lock

By default, a file is stored in a buffer before being written to archive media. If direct I/O is enabled, you can use the `-lock` parameter to lock this buffer. The `-lock` parameter indicates that the archiver must use locked buffers when making archive copies. If `-lock` is specified, the archiver sets file locks on the archive buffer in memory for the duration of the `sam-arcopy`(1M) operation. This action avoids paging of the buffer, and can improve performance.

This parameter has the following format:

```
-lock
```

Use the `-lock` parameter only on large systems with large amounts of memory. Insufficient memory can cause an out-of-memory condition.

The `-lock` parameter is effective only if direct I/O is enabled for the file. By default, `-lock` is not specified, and the file system sets locks on all direct I/O buffers, including those for archiving. For more information about enabling direct I/O, see the `setfa`(1) man page, the `sam_setfa`(3) library routine man page, or the `-O forcedirectio` option on the `mount_samfs`(1M) man page.

This paramter is similar to the `lock` argument to the `bufsize` global directive. For more information about this topic, see bufsize Directive: Setting the Archiver Buffer Size.

Making Archive Copies of Offline Files: -offline_copy

A file is a candidate for being released after one archive copy is made. If the file is released and goes offline before any remaining archive copies are made, the archiver uses this parameter to determine the method use to make the other archive copies. When you specify the method, consider the number of drives available to the SAM-QFS system and the amount of disk cache available. This parameter has the following format:

```
-offline_copy <method>
```

method Value	Meaning
`none`	Stages files as needed for each file before copying to the archive volume. Default.
`direct`	Copies files directly from the offline volume to the archive volume without using the cache. This method assumes that the source volume and the destination volume are different volumes and that two drives are available. Raise the value of the `stage_n_window` mount option to a value that is greater than its default of 256 kilobytes. For more information about mount options, see the `mount_samfs`(1M) man page.
`stageahead`	Stages the next archive file while writing an archive file to its destination.
`stageall`	Stages all files to disk cache before archiving. This method uses one drive and assumes that room is available on disk cache for all files.

Specifying Recycling

Use the recycling process to reclaim space on archive volumes in use by expired archive images. By default, no recycling occurs. You must specify directives in both the `archiver.cmd` file and the `recycler.cmd` file. For more information, see the Configuring the Recycler page.

Sorting Archive Files: -sort and -rsort

By default, files in an archive set are sorted by path before they are archived. You can specify that files are sorted by `age`, `priority`, `size` or not sorted (none). Only one sort method can be used per archive set.
You can use `-rsort` to reverse the order of sorting specified by method.

Example - Sorting Files in an Archive Set

The first example line sorts the archive set copy `cardiac.2` by the age of the file, oldest to youngest. The second line forces the archive set copy `catscans` to be sorted by the size of the file, in reverse order, largest to smallest.
`size`.

```
cardiac.2 -sort age
catscans.3 -rsort size
```

Controlling Unarchiving

Unarchiving is the process by which archive entries for files or directories are deleted. A file is unarchived based on the time since it was last accessed. This distinction means data that is accessed frequently can be stored on fast media such as disk and infrequently-accessed data can be stored on tape. By default, files are never unarchived.

Example - Directives to Control Unarchiving

This following example directives specify that the {{arset1}}file remains on disk all the time, even if it is older than 60 days. The Copy 1 information is removed when the file has not been accessed for 60 days. After the Copy 1 information is removed, any access request is fulfilled by Copy 2 and is read from tape. The archiver makes a new Copy 1 on disk and the 60-day cycle starts again.

```
arset1 dir1
1 10m 60d
2 10m
3 10m
vsns
arset1.1 mo OPT00[0-9]
arset1.2 lt DLTA0[0-9]
arset1.3 lt DLTB0[0-9]
```

The example directives meet the requirements for both access and archiving in the following scenario.
A patient is in the hospital for four weeks. During this time, all the patient's files are on fast media and the data is being access frequently. This is Copy 1 (copy 1=mo). After two weeks, the patient is discharged from the hospital. The patient files are accessed less frequently and then not at all. When no data has been accessed for this patient 60 days, the Copy 1 entry in the inode is unarchived. Only Copy 2 and Copy 3 entries are available. The volume of fast media can now be recycled and used by current patients without having to increase the disk library. However, six months later, the patient returns to the hospital. The first access of the patient's file is from tape (Copy 2). To get the data on fast media, the archiver creates a new Copy 1 on disk, ready for new information.

Controlling How Archive Files Are Written: -tapenonstop

By default, the archiver writes a tape mark, an end of file (EOF) label, and two more tape marks between archive files. When the next archive file is started, the driver backs up to the position after the first tape mark, causing a loss of performance. The `-tapenonstop` parameter directs the archiver to write only the initial tape mark. In addition,the archiver enters the archive information at the end of the copy operation.

For more information about the `-tapenonstop` parameter, see the `archiver.cmd`(4) man page.

Reserving Volumes: -reserve

By default, the archiver writes archive set copies to any volume specified by a regular expression as described in the volume associations section of the `archiver.cmd` file. However, if you require that an archive set volume contains files from only one archive set, you can reserve a volume for this purpose.

Note -

- A site that uses reserved volumes incurs more cartridge loads and unloads.
- A site that uses reserved volumes for file systems that have many directories of a few small files causes the archiver to write many small archive files to each reserved volume. These small archive files, each with its own `tar`(1) header, slow performance.

The `-reserve` parameter specifies a volume for use by an archive set and gives it a unique identifier that ties the archive set to the volume. The volume identifier is not assigned to any other archive set copy, even if a regular expression matches it. The format for the `-reserve` parameter is as follows:

```
-reserve <keyword>
```

The value of keyword depends on the form you are using. You can specify one, two, or all three forms in combination.

Form	keyword	Reserved Name Examples	Notes
Archive Set	`set`	`users.1//Data.1//`	

Owner	dir	proj.1/p105/ proj.1/p104/	The dir, user, and group keywords, which are mutually exclusive, specify the owner component in the reserved name. The dir keyword uses the directory path component immediately following the path specification of the archive set definition.
	user	users.1/user5/ users.1/user4/	
	group	data.1/engineering/	
File System	fs	proj.1/p103/samfs1 proj.1/p104/samfs1	The fs keyword specifies the file system component in the reserved name.

Example - Reserving Volumes by Archive Set

The following example specifies that the allsets archive set reserves a volume for each archive set.

```
params
allsets -reserve set
endparams
```

Example - Reserved Volume Name

The following example specifies that the arset.1 archive set reserves a volume and the volume identifier is created from an archive set, a group, and the file system.

```
params
arset.1 -reserve set -reserve group -reserve fs
endparams
```

Information about reserved volumes is stored in the library catalog. The lines in the library catalog that describe reserved volumes begin with #R characters and show the media type, the VSN, the reserve information, and the reservation date and time. The information also includes the archive set component, path name component, and file system component, separated by two slashes (//).

> **Note -**
> The slash characters do not indicate a path name. They serve to separate the components of a reserved name.

Example - Library Catalog Showing Reserved Volumes

The lines have been truncated to fit on the page.

```
6 00071 00071 lt 0xe8fe 12 9971464 1352412 0x6a000000 131072 0x
# -il-o-b----- 05/24/00 13:50:02 12/31/69 18:00:00 07/13/01 14:03:00
#R lt 00071 arset0.3// 2001/03/19 18:27:31
10 ST0001 NO_BAR_CODE lt 0x2741 9 9968052 8537448 0x68000000 1310
# -il-o------- 05/07/00 15:30:29 12/31/69 18:00:00 04/13/01 13:46:54
#R lt ST0001 hgm1.1// 2001/03/20 17:53:06
16 SLOT22 NO_BAR_CODE lt 0x76ba 6 9972252 9972252 0x68000000 1310
# -il-o------- 06/06/00 16:03:05 12/31/69 18:00:00 07/12/01 11:02:05
#R lt SLOT22 arset0.2// 2001/03/02 12:11:25
```

One or more of the reserve information fields can be empty, depending on the options defined in the archiver.cmd file. A reservation line is appended to the file for each volume that is reserved for an archive set during archiving.

You can also use the reserve(1M) and unreserve(1M) commands to reserve and unreserve volumes. For more information about these commands, see the reserve(1M) and unreserve(1M) man pages.
A volume is unreserved when it is relabeled because the archive data has been effectively erased.

You can display the reserve information by using the samu(1M) utility's v display or by using the archiver(1M) or dump_cat(1M) command in one of the formats shown in the following example.

Example - Commands to Use to Display the Reserve Information

```
archiver -lv
dump_cat -V _catalog-name_
```

Example: User and Data Files Archived to Optical Media shows a complete archive example using reserved volumes.

Setting Archive Priorities: -priority

Archive-enabled file systems provide priorities for archiving files. Each file is assigned a priority computed from properties of the file and priority multipliers that can be set for each archive set in the `archiver.cmd` file. Properties include online/offline, age, number of copies made, and size.

By default, the files in an archive request are not sorted, and all property multipliers are zero. The result is that files are archived in first-found, first-archived order. To change the order in which files are archived, set priorities and sort methods. Examples of new priorities include:

- Select the `priority` sort method to archive files within an archive request in priority order.
- Change the `archive_loaded` priority to reduce media loads.
- Change the `offline` priority to cause online files to be archived before offline files.
- Change the `copy#` priorities to make archive copies in copy order.

Table - Archive Priorities

Archive Priority	Definition
`-priority age` value	Archive age property multiplier
`-priority archive_immediate` value	Archive immediate property multiplier
`-priority archive_overflow` value	Multiple archive volumes property multiplier
`-priority archive_loaded` value	Archive volume loaded property multiplier
`-priority copies` value	Copies-made property multiplier
`-priority copy1` value	Copy 1 property multiplier
`-priority copy2` value	Copy 2 property multiplier
`-priority copy3` value	Copy 3 property multiplier
`-priority copy4` value	Copy 4 property multiplier
`-priority offline` value	File offline property multiplier
`-priority queuewait` value	Queue wait property multiplier
`-priority rearchive` value	Rearchive property multiplier
`-priority reqrelease` value	Reqrelease property multiplier
`-priority size` value	File-size property multiplier
`-priority stage_loaded` value	Stage volume loaded property multiplier
`-priority stage_overflow` value	Multiple stage volumes property multiplier

For value, specify a floating-point number in the following range:

```
-3.400000000E+38 <= _value_ <= 3.402823466E+38
```

For more information about priorities, see the `archiver`(1M) and `archiver.cmd`(4) man pages.

Scheduling Archiving: -startage, -startcount, and -startsize

As the archiver scans a file system, it identifies files to be archived. Files that are recognized as candidates for archiving are placed in a list known as an archive request. At the end of the file system scan, the system schedules the archive request for archiving. The `-startage`, `-startcount`, and `-startsize` archive set parameters control the archiving workload and ensure the timely archival of files.

Table - Formats for the **`-startage`**, **`-startcount`**, and **`-startsize`** Directives

Directive	Meaning
`-startage` time	The amount of time that can elapse between the first file in a scan being marked for inclusion in an archive request and the start of archiving. Specify a time in the format used in Setting the Archive Age. If this variable is not set, the `interval` directive is used.
`-startcount` count	The number of files to be included in an archive request. When the number of files in the archive request reaches the this value, archiving begins. By default, count is not set.
`-startsize` size	The minimum total size, in bytes, of all files to be archived in an archive request. Archiving work is accumulated, and archiving begins when the total size of the files reaches the this value. By default, size is not set.

The `examine=`method directive and the `interval=`time directives interact with the `-startage`, `-startcount`, and `-startsize` directives. The `-startage`, `-startcount`, and `-startsize` directives optimally balance archive timeliness and archive work done. These values override the `examine=`method specification, if any. For more information about `examine` directive, see examine Directive: Controlling Archive Scans. For more information about the `interval` directive, see interval Directive: Specifying an Archive Interval.

The `-startage`, `-startcount`, and `-startsize` directives can be specified for each archive copy. If more than one of these directives is specified, the first condition encountered starts the archive operation. If none of these directives is specified, the archive request is scheduled based on the `examine=`method directive:

- If `examine=noscan`, the default values of the directives are used: `startage` 10 minutes, `startcount` 10,000, and `startsize` 10 gigabytes. The archive request is scheduled according to the value of the `interval=` directive after the first file is entered in the archive request. This method is continuous archiving and is the default method.
- If `examine=scan|scaninodes|scandirs`, the archive request is scheduled for archiving after the file system scan.

The `archiver.cmd`(4) man page provides examples that show how to use these directives.

VSN Association Directives

The VSN associations section of the `archiver.cmd` file assigns volumes to archive sets. This section starts with a `vsns` directive and ends with an `endvsns` directive.

VSN associations can also be configured with the SAM-QFS Manager software. See the SAM-QFS Manager online help for more information.

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

```
<archive-set-name>.<copy-num> <media-type> <vsn-expr> ... [-pool <vsn-pool-name> ...]
```

An association requires at least three fields: archive-set-name and copy-num, media-type, and at least one volume. The archive-set-name and copy-num values are connected by a period (.).

Table - Arguments for the VSN Association Directive

Argument	Meaning
archive-set-name	A site-defined name for the archive set.
copy-num	A digit followed by one or more arguments that specify archive characteristics for that copy. Archive copy directives begin with a digit. This digit (1, 2, 3, or 4) is the copy number.
media-type	The media type. For a list of valid media types, see the `mcf`(4) man page.
vsn-expr	A regular expression. See the `regexp`(5) man page.
`-pool` vsn-pool-name	A named collection of VSNs.

> **Note -**
> If your SAM-QFS environment is configured to recycle by archive set, do not assign a VSN to more than one archive set.

Example - VSN Specifications on Multiple Lines

The following example shows two lines of VSN specifications.

```
vsns
set.1 lt VSN001 VSN002 VSN003 VSN004 VSN005
set.1 lt VSN006 VSN007 VSN008 VSN009 VSN010
endvsns
```

Example - VSN Specifications With a Continued Line

The following example shows a VSN specification that uses a backslash character "\" to continue a line onto a subsequent line.

```
vsns
set.1 lt VSN001 VSN002 VSN003 VSN004 VSN005 \
VSN006 VSN007 VSN008 VSN009 VSN010
endvsns
```

Example - VSN Specifications With Shorthand Notation

The following example specifies VSNs using a regular expression in a shorthand notation.

```
vsns
set.1 lt VSN0[1-9] VSN10
endvsns
```

When the archiver needs volumes for the archive set, it examines each volume of the selected media type in all automated libraries and manually mounted drives to determine whether the volume satisfies any VSN expression. It selects the first volume that matches an expression that contains enough space for the archive copy operation. For example:

- The following directive specifies that files belonging to archive set `ex_set` for copy 1 be copied to media type `mo` using any of the 20 volumes with the names `optic20` through `optic39`.

  ```
  ex_set.1 mo optic[2-3][0-9]
  ```

- The following directive specifies that files belonging to archive set `ex_set` for copy 2 be copied to media type `lt` using any volume beginning with TAPE:

  ```
  ex_set.2 lt ^TAPE
  ```

  > **Note -**
  > Make sure you assign volumes to the archive set used for the file system's metadata when setting up the `archiver.cmd` file. Each file system has an archive set with the same name as the file system. For more information about preserving metadata, see the `samfsdump`(1M) man page or see the SAM-QFS Troubleshooting page.

VSN Pools Directives

The VSN pools section of the `archiver.cmd` file starts with a `vsnpools` directive and ends either with an `endvsnpools` directive or with the end of the `archiver.cmd` file. This section names a collection of volumes.

VSN pools can also be configured with the SAM-QFS Manager software. See the SAM-QFS Manager online help for more information.

A VSN pool is a named collection of volumes. VSN pools are useful for defining volumes that can be available to an archive set. As such, VSN pools provide a useful buffer for assigning volumes and reserving volumes to archive sets. You can use VSN pools to define separate groups of volumes by departments within an organization, by users within a group, by data type, and according to other convenient groupings.

If a volume is reserved, it is no longer available to the pool in which it originated. Therefore, the number of volumes within a named pool changes as volumes are used. You can view the VSN pools by issuing the `archiver`(1M) command in the following format:

```
# archiver -lv | more
```

The syntax of a VSN pool definition is as follows:

```
<vsn-pool-name> <media-type> <vsn-expr>
```

Argument	Meaning
vsn-pool-name	The VSN pool.
media-type	The two-character media type. For a list of valid media types, see the `mcf`(4) man page.
vsn-expr	A regular expression. You can provide one or more vsn-expr arguments. See the `regcmp`(3G) man page.

The following example uses four VSN pools: `users_pool`, `data_pool`, `proj_pool`, and `scratch_pool`. A scratch pool is a set of volumes used when specific volumes in a VSN association are exhausted or when another VSN pool is exhausted. If one of the three specific pools is out of volumes, the archiver selects the scratch pool VSNs.

Example - VSN Pools

The following example shows an `archiver.cmd` file that uses four VSN pools.

```
vsnpools
users_pool mo ^MO[0-9][0-9]
data_pool mo ^DA.*
scratch_pool mo ^SC[5-9][0-9]
proj_pool mo ^PR.*
endvsnpools
vsns
users.1 mo -pool users_pool -pool scratch_pool
data.1 mo -pool data_pool -pool scratch_pool
proj.1 mo -pool proj_pool -pool scratch_pool
endvsns
```

For more information about VSN associations, VSN Association Directives

Checking the Drive Order in Libraries

Contents

Checking the Drive Order in Libraries

If your automated library contains more than one drive, the order of the drives in the `mcf` file must be the same as the order of the drives displayed by the automated library's controller. The drive that the library controller identifies as the first drive must be the first drive entry for that library in the `mcf` file, and so on. This order can be different from the order of the devices as reported in the `/var/adm/messages` file.

The following sections describe procedures for checking the drive order. The procedure varies depending on whether your automated library has a front panel, whether it has tape or magneto-optical drives, and whether it is direct attached or network attached. Each procedure has two phases:

- Mapping the library drives to SCSI target IDs
- Mapping the SCSI target IDs to remote tape devices

How to Check the Drive Order of Libraries With a Front Panel

Some libraries have a panel that display the drive information.
The following procedure is a general plan. The actual steps depend on your specific library product. Consult your vendor documentation for information about drive identification and target identification.

1. Verify the order of the drives, according to the vendor's documentation.
2. In the front panel, check each drive's SCSI target ID or World Wide Name (WWN).
3. Record the order in which each drive and drive target is reported.
4. In the `mcf` file, make sure that the order of the drive targets is in the same order in which the drives are displayed by the automated library's controller.
5. If you made any changes, verify the `mcf` file and test the drives. Then propagate the change to the rest of the system. For information about propagating `mcf` file changes, see the Sun QFS File System Configuration and Administration Guide.

To determine whether the drives become active when loaded with a cartridge, you can visually inspect the drives or use the `samu`(1M) utility's `r` display.

How to Check the Drive Order of a Tape Library Without a Front Panel

1. Stop the SAM software so that no drives are used during the procedure.
2. Get a listing of devices in /dev/rmt/, as shown in the following example:

```
# ls -l /dev/rmt/?
lrwxrwxrwx 1 root root 42 Jan 10 2000 /dev/rmt/0 ->
../../devices/pci@1f,4000/scsi@2,1/st@2,0:
lrwxrwxrwx 1 root root 42 Jan 10 2000 /dev/rmt/1 ->
../../devices/pci@1f,4000/scsi@4,1/st@5,0:
lrwxrwxrwx 1 root root 42 Jan 10 2000 /dev/rmt/2 ->
../../devices/pci@1f,4000/scsi@4,1/st@6,0:
lrwxrwxrwx 1 root other 40 Dec 13 2000/dev/rmt/3 ->
../../devices/pci@1f,4000/scsi@4/st@1,0:
lrwxrwxrwx 1 root root 40 Jun 20 2001 /dev/rmt/4 ->
../../devices/pci@1f,4000/scsi@4/st@2,0:
lrwxrwxrwx 1 root root 40 Jun 20 2001 /dev/rmt/5 ->
../../devices/pci@1f,4000/scsi@4/st@3,0:
lrwxrwxrwx 1 root root 40 Jun 20 2001 /dev/rmt/6 ->
../../devices/pci@1f,4000/scsi@4/st@4,0:
lrwxrwxrwx 1 root root 40 Sep 14 2001 /dev/rmt/7 ->
../../devices/pci@1f,4000/scsi@2/st@2,0:
lrwxrwxrwx 1 root root 40 Sep 14 2001 /dev/rmt/8 ->
../../devices/pci@1f,4000/scsi@2/st@3,0:
lrwxrwxrwx 1 root root 40 Sep 14 2001 /dev/rmt/9 ->
../../devices/pci@1f,4000/scsi@2/st@4,0:
```

3. Load a tape into the library's Drive 1. Make sure the other drives are empty.

```
samload
```

4. Run the following command with each /dev/rmt/ entry until the command returns information about the drive and tape position:

```
mt -f /dev/rmt/<x> status
```

   The /dev/rmt/# entry that returns information corresponds to the library's Drive 1. The following example shows `mt`(1) command output that indicates that 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
```

5. Repeat the steps for each library drive.
   Create a table that shows which library drive corresponds to which /dev/rmt/ entry:

```
   drive 1 = /dev/rmt/4 -> ../../devices/pci@1f,4000/scsi@4/st@2,0:
   drive 2 = /dev/rmt/7 -> ../../devices/pci@1f,4000/scsi@2/st@2,0:
   ...
```

6. In the `mcf` file, make sure that the drives are listed in the order in which the drives are displayed by the automated library's controller.
   In this case, the `mcf` file starts with the following items:

```
   # Equipment          Eq   Eq    Family   Device   Additional
   # Identifier         Ord Type    Set     State    Parameters
   #-----------         --- ----   ------   ------   ----------
   /dev/rmt/4           31   li    ibm3580   on
   /dev/rmt/7           32   li    ibm3580   on
    ...
```

7. If you made any changes, verify the `mcf` file and test the drives. Then propagate the change to the rest of the system. For information about propagating `mcf` file changes, see the Sun QFS File System Configuration and Administration Guide.

How to Check the Drive Order of Magneto-Optical Libraries Without a Front Panel

1. Stop the SAM software so that no drives are used during the procedure.
2. Get a listing of devices in /dev/samst/, as shown in the following example:

```
# ls -l /dev/samst/?
```

3. Load a magneto-optical cartridge manually through the library front panel into the library's Drive 1. Make sure the other drives are empty.
4. Run the following command with each /dev/samst/ entry until the command returns information about the drive and tape position:

```
dd if=</dev/samst/x> bs=2k iseek=3374 of=/tmp/foo count=10
```

   The /dev/samst/ entry that returns information corresponds to the library's Drive 1. The following example shows a status message that indicates that an optical cartridge is in the selected device.

```
# dd if=/dev/samst/c0t3u0 bs=2k iseek=3374 of=/tmp/junk count=10
10+0 records in
10+0 records out
```

5. Repeat the steps for each library drive.
   Create a table that shows which library drive corresponds to which /dev/samst/ entry:

```
   drive 1 = /dev/samst/4 -> ../../devices/pci@1f,4000/scsi@4/st@2,0:
   drive 2 = /dev/samst/7 -> ../../devices/pci@1f,4000/scsi@2/st@2,0:
   ...
```

6. In the `mcf` file, make sure that the drives are listed in the order in which the drives are displayed by the automated library's controller.
   In this case, the `mcf` file starts with the following items:

```
# Equipment         Eq   Eq    Family   Device   Additional
# Identifier        Ord Type    Set     State    Parameters
#-----------        --- ----   ------   ------   ----------
/dev/samst/4          31   li    ibm3580   on
/dev/samst/7          32   li    ibm3580   on
 ...
```

7. If you made any changes, verify the mcf file and test the drives. Then propagate the change to the rest of the system. For information about propagating mcf file changes, see the Sun QFS File System Configuration and Administration Guide.

How to Check the Drive Order of Network Attached Libraries

1. Stop the SAM software so that no drives are used during the procedure.
2. Get a listing of devices in /dev/rmt/, as shown in the following example:

```
# ls -l /dev/rmt/*[0-9] | awk '{print $9, $10, $11}'
/dev/rmt/0 -> /devices/pci@8,700000/SUNW,qlc@4,1/fp@0,0/st@w500104f0006041f0,0:
/dev/rmt/1 -> /devices/pci@8,700000/SUNW,qlc@4,1/fp@0,0/st@w500104f0006041f3,0:
/dev/rmt/2 -> /devices/pci@8,700000/SUNW,qlc@4,1/fp@0,0/st@w500104f00043cbb8,0:
/dev/rmt/3 -> /devices/pci@8,700000/SUNW,qlc@5,1/fp@0,0/st@w500104f0006041ea,0:
/dev/rmt/4 -> /devices/pci@8,700000/SUNW,qlc@5,1/fp@0,0/st@w500104f0006041ed,0:
/dev/rmt/5 -> /devices/pci@8,700000/SUNW,qlc@4/fp@0,0/st@w500104f00060420e,0:
/dev/rmt/6 -> /devices/pci@8,700000/SUNW,qlc@4/fp@0,0/st@w500104f000604211,0:
/dev/rmt/7 -> /devices/pci@8,700000/SUNW,qlc@4/fp@0,0/st@w500104f000604214,0:
/dev/rmt/8 -> /devices/pci@8,700000/SUNW,qlc@5/fp@0,0/st@w500104f000604208,0:
/dev/rmt/9 -> /devices/pci@8,700000/SUNW,qlc@5/fp@0,0/st@w500104f00060420b,0:
```

   Next we can use luxadm output, along with output from an ACSLS display command and associate the serial numbers of each drive with the physical location in the library.

3. For each device, use the luxadm display command to show its serial number:

```
# luxadm display /dev/rmt/<x>
```

4. Use the ACSLS display to display the drive identifier for each serial number.

```
ACSSA> display drive * -f serial_num
2007-10-11 10:49:12             Display Drive
Acs  Lsm  Panel  Drive  Serial_num
0    2    10     12      331000049255
0    2    10     13      331002044567
0    2    10     14      331002057108
0    2    10     15      331002042417
0    2    10     16      331002031352
0    2    10     17      HU92K00200
0    2    10     18      HU92K00208
0    3    10     10      1200019405
0    3    10     11      1200019442
0    3    10     12      1110150718
```

5. Create a table to show the relationships among the identifiers:

```
 Device          SSN             Drive Identifier
/dev/rmt/0 -> 331000049255 -> (acs=0, lsm=2, panel=10, drive=12)
/dev/rmt/1 -> 331002044567 -> (acs=0, lsm=2, panel=10, drive=13)
/dev/rmt/2 -> 331002057108 -> (acs=0, lsm=2, panel=10, drive=14)
```

6. In the mcf file, verify that the order of the drives is in the same order as the table.
7. If you made any changes, verify the mcf file and test the drives. Then propagate the change to the rest of the system. For information about propagating mcf file changes, see the Sun QFS File System Configuration and Administration Guide.

Configuring Storage Devices for Archiving

Contents

Configuring Storage Devices for Archiving

Perform the tasks in this section if you plan to enable archiving to tape or magneto-optical media. If you plan to archive to disk, you do not need to perform these tasks.

This section describes how to verify and update the following files:

- `/kernel/drv/st.conf`, which configures tape drives attached to the server through a SCSI or FC attachment.
- `/kernel/drv/samst.conf`, which configures the following devices that the Sun Storage Archive Manager (SAM) software recognizes by default:
  - Direct attached automated libraries
  - Magneto-optical drives attached to the server through a SCSI or FC attachment

The SAM package includes the `/opt/SUNWsamfs/examples/st.conf_changes` file, which contains configuration information for tape drives that are not supported in the Solaris kernel by default.

Task Map: Configuring Storage Devices for Archiving

Depending on the devices you want to configure, you must complete one or more of the following procedures.

Step	Task	Description
1	Create List of Devices	Take an inventory of your devices to configure.
2	Add Tape Devices	Perform this task for each tape drive that you want to add to the SAM environment.
3	Add Tape Drive Interface Target IDs and LUNs	Perform this task if your tape drives are attached through a SCSI or FC interface.
4	Add Libraries or Magneto-Optical Drives	Perform this task if you have any magneto-optical drives, SCSI-attached automated libraries, or FC-attached automated libraries that you want to include in your SAM environment.
5	Verify Configured Devices	Verify that you have all of your devices configured properly.
6	Enable the Storage Device Configuration	Reboot the system to enable the changes you have made.
7	Create Parameters Files for Network Attached Automated Libraries	Create parameter files for the network attached automated libraries.

The procedures in this task include an example that is based on the inventory list shown in the following table.

Example Inventory List - Devices to Be Configured

Device Name, Manufacturer, and Model	Target ID	LUN	Node WWN
SCSI-attached tape drives			

QUANTUM DLT7000	1	0	Not applicable
QUANTUM DLT7000	2	0	Not applicable
FC-attached tape drives			
StorageTek 9840	Not applicable	0	500104f00043abfc
StorageTek 9840	Not applicable	0	500104f00045eeaf
IBM ULT3580-TD1	Not applicable	0	500104f000416304
IBM ULT3580-TD1	Not applicable	0	500104f000416303
SCSI-attached automated libraries			
StorageTek 9730	0	0	Not applicable
FC-attached automated libraries			
StorageTek L700	Not applicable	0	500104f00041182b

Note -
The device names are shown as they appear in the discovery output.

How To Add Tape Devices for Archiving

Perform this procedure to include your tape drives in your archiving environment. In this procedure, you make an entry in the `st.conf` file for each unique tape drive that is on your inventory list.

Tip -
You can also add tape devices from SAM-QFS Manager. For information, see How to Add Tape Devices From SAM-QFS Manager.

1. Copy `/kernel/drv/st.conf` to a backup file. For example:

```
# cp /kernel/drv/st.conf /kernel/drv/st.conf.orig
```

2. Open the `/kernel/drv/st.conf` file and find the following string:

```
#tape-config-list=
```

3. Remove the pound character (#).
4. Open the `/opt/SUNWsamfs/examples/st.conf_changes` file. For each tape drive on your inventory list, do the following.
   a. Search for the device definition entry.
      For example, searching for the Quantum DLT 7000 tape drive that is in the example inventory locates the following entry:

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

   b. Copy the entry from the `st.conf_changes` file to the `st.conf` file, placing it after the `tape-config-list` line. The following example shows the resulting `st.conf` file.

```
...
tape-config-list=
"QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape",
...
```

   c. Note the final string in the entry, which is enclosed in quotation marks. In this example, the final string is "`dlt7-tape`". This string is the tape configuration value.

d. Search /opt/SUNWsamfs/examples/st.conf_changes to find the line that begins with this tape configuration value. In this example, the value is:

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

e. Copy the tape configuration value to the st.conf file, placing it after the device definition line. The following example shows the lines now contained in the st.conf file.

```
...
tape-config-list=
"QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape";
dlt7-tape =    1,0x36,0,0xd679,4,0x82,0x83,0x84,0x85,3;
...
```

f. Repeat this procedure for each type of tape drive.

> Note -
> In the st.conf_changes file, a tape configuration value is repeated for each device definition that uses the same tape configuration. In the st.conf file, include only one entry for each tape configuration value. For example, the Sony SDT-5000 and the Sony SDT-5200 tape drives both use "DAT" as the final string. A single entry for the DAT tape configuration value is sufficient.

5. Replace the comma (,) at the end of the last device definition line with a semicolon (;).
   The following example shows a st.conf file with definitions for the Quantum DLT 7000, the StorageTek 9840, and the IBM ULT3580 tape drives. The semicolon is after "CLASS_3580"

```
...
tape-config-list=
"QUANTUM DLT7000", "DLT 7000 tape drive", "dlt7-tape",
"STK      9840", "STK 9840 Fast Access", "CLASS_9840",
"IBM      ULT3580-TD1",  "IBM 3580 Ultrium",      "CLASS_3580";
dlt7-tape =    1,0x36,0,0xd679,4,0x82,0x83,0x84,0x85,3;
CLASS_9840 = 1,0x36,0,0x1d679,1,0x00,0;
CLASS_3580 = 1,0x24,0,0x418679,2,0x00,0x01,0;
...
```

6. Save your changes.
7. Close the /opt/SUNWsamfs/examples/st.conf_changes file.
   Do not close the st.conf file because you continue to edit the file in the next procedure.
8. Go to the #How To Add Tape Drive Interface Target IDs and LUNs for Archiving procedure.

How To Add Tape Drive Interface Target IDs and LUNs for Archiving

For each tape drive on your hardware inventory list that is attached through a SCSI or FC interface, you must confirm that an entry in the st.conf file defines that interface. This procedure shows how to verify and, if necessary, add target ID and LUN entries.

> Note -
> Do not use this procedure to add interface information for magneto-optical drives. See How To Add Libraries or Magneto-Optical Drives for Archiving

1. Open the file /kernel/drv/st.conf if it is not already open.
2. To use the SCSI interface to attach tape drives, do the following:
   a. Search for entries that have the following format to locate the list of SCSI target IDs and LUNs:

```
name="st" class="scsi" target=<target> lun=<lun>;
```

target is the target ID for each SCSI drive found.
lun is the corresponding LUN for each SCSI drive found.

b. Within the list, find each entry that corresponds to each SCSI target and LUN on your hardware inventory list. The following example shows the two entries that correspond to the two Quantum DLT 7000 drives that are attached to LUN 0 and have target IDs 1 and 2, shown in Example Inventory List.

```
name="st" class="scsi" target=1 lun=0;
name="st" class="scsi" target=2 lun=0;
```

Note that an entry might extend over two lines.

- If the entry is preceded by a pound character (#), delete the character. A pound character marks a line as a comment.
- If the entry is missing, create an entry for the SCSI target and LUN line you need, following the format shown in Step a and using the information in your hardware inventory list.

3. To use the FC interface to add tape drives and if you are not using the Sun StorageTek SAN Foundation Software I/O stack, create a line for each FC-attached device after the SCSI target ID and LUN list, using the following format:

```
name="st" parent="fp" lun=<lun> fc-port-wwn="<world-wide-name>"
```

For lun, specify the LUN for the drive.
For world-wide-name, specify the World Wide Name (WWN) for the drive.
The following example shows the lines that support the StorageTek 9840 and IBM ULT3580 tape drives included in the Example Inventory List.

```
name="st" parent="fp" lun=0 fc-port-wwn="500104f00043abfc"
name="st" parent="fp" lun=0 fc-port-wwn="500104f00045eeaf"
name="st" parent="fp" lun=0 fc-port-wwn="500104f000416304"
name="st" parent="fp" lun=0 fc-port-wwn="500104f000416303"
```

4. Save your changes and close the `st.conf` file.
5. To include magneto-optical drives or automated libraries attached through a SCSI or a Fibre Channel interface, go to the How To Add Libraries or Magneto-Optical Drives for Archiving procedure.

How to Add Tape Devices From SAM-QFS Manager

To add a tape device from the SAM-QFS Manager:

1. In the left pane, click Archive Media.
2. In the Tape Library Summary window, click Add.
3. Follow the steps in the wizard.

How To Add Libraries or Magneto-Optical Drives for Archiving

The `inquiry.conf` file lists all devices that are supported.
The `/kernel/drv/samst.conf` file contains a list of SCSI and FC entries and works with the `/opt/SUNWsamfs/examples/inquiry.conf` file to define the devices that can be included in a SAM environment.
The following procedures show how to verify the entries in the `samst.conf` and to update the file if necessary.

You update the `samst.conf` file, depending on your environment:

- If you have only network-attached automated libraries, you do not need to verify device support or update it.
- If you use a SCSI or FC interface to attach a tape library to the server, use the procedure in How To Configure Device Support in SCSI or FC Environments to discover the tape libraries with the current drive information.
- If you have a direct attached library with a target number larger than 6 or a LUN identifier larger than 1, use the procedure in How To Configure Device Support for a Direct Attached Library.

How To Configure Device Support in SCSI or FC Environments

Repeat this procedure for each device in your environment.

1. On the SAM-QFS Manager Managed Hosts page, select the name of the server to which you want to add a library.
   The File Systems Summary page is displayed.
2. Expand the Archive Media section and select Tape Libraries.
   The Tape Library Summary page is displayed.
3. Click Add to start the Add a Library wizard.
4. Follow the steps to add the device.
   When you have completed the wizard steps, the `samst.conf` file is automatically updated.

How To Configure Device Support for a Direct Attached Library

1. Copy the `/kernel/drv/samst.conf` file to a backup file. For example:

```
# cp /kernel/drv/samst.conf /kernel/drv/samst.conf.orig
```

2. Edit the `/kernel/drv/samst.conf` file.
3. To include SCSI-attached magneto-optical drives or SCSI-attached libraries, do the following:
   a. Search for entries that have the following format to locate the list of SCSI targets and LUNs:

   ```
   name="samst" class="scsi" target=<target> lun=<lun>;
   ```

   target is the target ID for each SCSI drive found.
   lun is the corresponding LUN for each SCSI drive found.

   b. Within the list, find the entry that corresponds to each SCSI target ID and LUN on your inventory list.
   For example, the StorageTek 9730 automated library is attached to target 0 and LUN 0. The following line corresponds to that interface:

   ```
   name="samst" class="scsi" target=0 lun=0;
   ```

   Note that an entry might extend over two lines if it contains return characters.

   - If the entry starts with a pound character (#), delete the character. A pound (#) character marks a line as a comment.
   - If the entry is missing, create an entry for the SCSI target and LUN line you need, following the format shown in Step a and using the information in your hardware inventory list.
4. To include FC-attached magneto-optical drives or FC-attached automated libraries, create a line for each FC-attached one in your inventory list. Place the lines at the end of the SCSI target and LUN list, using the following format:

```
name="samst" parent="fp" lun=<lun> fc-port-wwn="<world-wide-name>"
```

For lun, specify the LUN for the drive.
For world-wide-name, specify the WWN for the drive.
The following example shows the line added to support the StorageTek L700 tape drive in the Example Inventory List.

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

5. Save your changes and exit the `samst.conf` file.

Verifying That All Devices Are Configured

1. Use the `cfgadm`(1M) command to list the devices included in the SAM environment. For example:

```
# cfgadm -al
Ap_Id                          Type         Receptacle   Occupant     Condition
c0                             scsi-bus     connected    configured   unknown
c0::dsk/c0t6d0                 CD-ROM       connected    configured   unknown
c1                             fc-private   connected    configured   unknown
c1::500000e0103c3a91           disk         connected    configured   unknown
c2                             scsi-bus     connected    unconfigured unknown
c3                             scsi-bus     connected    unconfigured unknown
c4                             scsi-bus     connected    configured   unknown
c4::dsk/c4t1d0                 disk         connected    configured   unknown
c4::dsk/c4t2d0                 disk         connected    configured   unknown
c5                             fc-fabric    connected    configured   unknown
c5::100000e00222ba0b           disk         connected    unconfigured unknown
c5::210000e08b0462e6           unknown      connected    unconfigured unknown
c5::210100e08b2466e6           unknown      connected    unconfigured unknown
c5::210100e08b27234f           unknown      connected    unconfigured unknown
c5::500104f00043abfc           tape         connected    configured   unknown
c5::500104f00043bc94           tape         connected    configured   unknown
c5::500104f00045eeaf           tape         connected    configured   unknown
c5::500104f000466943           tape         connected    configured   unknown
c5::500104f00046b3d4           tape         connected    configured   unknown
c5::500104f0004738eb           tape         connected    configured   unknown
c6                             fc           connected    unconfigured unknown
c7                             scsi-bus     connected    unconfigured unknown
c8                             scsi-bus     connected    unconfigured unknown
usb0/1                         usb-kbd      connected    configured   ok
usb0/2                         usb-mouse    connected    configured   ok
usb0/3                         unknown      empty        unconfigured ok
usb0/4                         unknown      empty        unconfigured ok
```

2. Examine the output to make sure that it shows all the devices you want configured in your SAM environment.
   If a device is not configured, use the `cfgadm`(1M) command to configure it. For more information, see the `cfgadm`(1M) man page.
   Because of a bug in the `cfgadm`(1) command, you might receive an error similar to the following:

```
# cfgadm -c configure -o force_update c4::500104f000489fe3
cfgadm: Library error: failed to create device node: 500104f00043abfc: Device busy
```

   Despite the error, the `cfgadm`(1M) command processes the request.

Enable Storage Device Configuration

You must reboot the system to have the changes you have made to the `st.conf` and `samst.conf` files take effect.

```
# reboot -- -r
```

Handling Errors in the `st.conf` File

Errors can occur if the `st.conf` file is not configured properly during SAM software installation.

The following messages in the `sam-log` file indicate that the appropriate changes have not been made to `/kernel/drv/st.conf`. Follow the steps in How To Add Tape Devices to the `/kernel/drv/st.conf` File to fix the error.

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

The following device log messages correspond to the `sam-log` message:

```
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
```

Configuring the Archiver

Contents

Configuring the Archiver

This page shows how to configure the archiver by modifying the `archiver.cmd file`.

About the **`archiver.cmd`** File

The `archiver.cmd` file controls the archiver's behavior.

The archiver copies files from a file system to volumes on removable media cartridges or to disk partitions in another file system. You can tune the archiver operations to suit the types of files at your site and to suit your site's data protection needs by creating an archiver command file named `/etc/opt/SUNWsamfs/archiver.cmd`. You are not required to create an `archiver.cmd` file, but the efficiency and performance of the archiver is improved if you tune the archiver to your site.

By default, the archiver starts whenever the `sam-fsd` process is started and a file system is mounted.
If the `archiver.cmd` file does not exist, the archiver is put in a `wait` state. To restart the archiver, use the `samcmd arrun` command. If no `archiver.cmd` file is found after the restart, the archiver continues to run, using the following default settings:

- Archives all files to all available volumes in all configured libraries.
- Makes one copy of each file.
- Sets the archive age for all files to 4 minutes. The archive age is the amount of time since a file's last modification.
- Sets the archive interval to 10 minutes. The archive interval is the amount of time that elapses between complete archiving processes.

To tune the actions of the archiver for your site, set directives in the `archiver.cmd`. A directive acts like a command parameter and consist of lines of text in the `archiver.cmd` file. The following rules apply to the `archiver.cmd` file:

- Each directive line contains one or more fields separated by spaces or tabs.

- Any text that appears after the pound sign character (#) is treated as a comment and is not examined.
- Lines that end with a backslash character () are joined to the next line.
  The `archiver.cmd` file uses two types of directives: Archive Directives and Archive Set Directives.
  For comprehensive information about the `archiver.cmd` directives, see the `archiver.cmd`(4) man page.

Whenever you make changes to the `archiver.cmd` file, check for syntax errors using the `archiver`(1M) command. This command produces a list of all options and writes a list of the volumes, file system content, and errors to the standard output file (`stdout`). If you encounter errors, correct them in the file and run the `archiver`(1M) command again to verify your corrections. The archiver does not archive any files if it finds errors in the `archiver.cmd` file.

Archive Directives

Archive directives specify the general archive operations and consist of two main areas in the `archiver.cmd` file:

- At the top of the file, global directives affect all file systems defined in your `mcf` file.
- The lower part of the file contains file-system-specific directives. These directives must come after the global directives. For any file system, these directives override the global directives. The file-system-specific directives start with an `fs=`name directive that identifies the file system.
  Within a file-system's directives, you can also specify archive copy directives to customize the number and frequency of the archive copies.

See the Archive Directives page for detailed information.

Archive Set Directives

An archive set identifies a group of files to be archived, regardless of their file system. A file in a file system can be a member of only one archive set. Files in an archive set share common criteria that pertain to the size, ownership, group, or directory location.
The archive set controls the destination of the archive copy, how long the copy is kept archived, and how long the software waits before archiving the data. All files in an archive set are copied to the volumes associated with that archive set.
The directives for an archive set include:

- Assignment directives define archive sets.
- Copy parameters define how each archive set is archived.
- VSN association directives assign volumes to archive sets.
- VSN pools directives define a collection of volumes.

See the Archive Set Directives page for detailed information.

How to Start the Archiver

The archiver starts automatically when a file system is mounted.
If the `archiver.cmd` file does not exist, the archiver is put in a `wait` state.

To restart the archiver, use the `samcmd arrun` command.
If the `archiver.cmd` file does not exist, the archiver runs in its default manner, as described in About the archiver.cmd File.

How to Create an **`archiver.cmd`** File Using the Command Line

- If your site has an `/etc/opt/SUNWsamfs/archiver.cmd` file and your system is already archiving files, do not make changes to it. Copy the file to a location where you can edit and test it. When it is verified, replace the existing file with the new one.
- If your site does not have a `archiver.cmd` file, you can edit the file in the `/etc/opt/SUNWsamfs` directory.

1. Edit the `archiver.cmd` file to add or change the directives that control archiving at your site. For information about the directives you can include in this file, see Archive Directives and Archive Set Directives for details.
2. Save and close the `archiver.cmd` file.
3. Verify the file for the current SAM-QFS environment. If you are working with a test `archiver.cmd` file, use the `-c` option with the `archiver`(1M) command and supply the file name.

```
# archiver -lv
```

4. If you are working with a test file, move it to `/etc/opt/SUNWsamfs/archiver.cmd`.
5. Use the `samd config` command to propagate the file changes and restart the system.

```
# samd config
```

How to Create an **archiver.cmd** File Using SAM-QFS Manager

When you create or edit an archive policy for a file system within the SAM-QFS Manager interface, the archiver.cmd file is automatically created or edited.

1. On the Servers page, select the name of the server for which you want to create a policy.
   The File Systems Summary page is displayed.
2. Select the Archive Administration node in the navigation tree.
   The Archive Policies Summary page is displayed.
3. Click New.
   The New Archive Policy wizard is displayed.
4. Follow the steps in the wizard.
   For detailed instructions on using the New Archive Policy wizard, see the SAM-QFS Manager online help.
   When you save the new archive policy, it is automatically written to the archiver.cmd file.

Example archiver.cmd Files

This section provides some examples of archiving configurations.

Example - Simple archiver.cmd File

The following example shows a simple archiver.cmd file that you can modify. Add directives only to accommodate more archive sets, copies, and VSN usage.

```
# archiver.cmd
# One file system = samfs1

archmax = sg 2G
examine = noscan

fs = samfs1
logfile = /var/opt/SUNWsamfs/log/samfs1.log
all_archset .
  1 -norelease 10m
  2 -norelease 10m

params
allsets -sort path -offline_copy stageahead -reserve set
allsets -recycle_hwm 50 -recycle_mingain 90 -recycle_vsncount 5 -recycle_dataquantity 40G
allsets.1 -startage 6h -startsize 6G -startcount 30000
allsets.2 -startage 10h -startsize 12G -startcount 60000 -archmax 12G
endparams

vsns
all.1 li .*
all.2 li .*
endvsns
```

Example - Advanced archiver.cmd File

The following example shows a complex archiver.cmd file. The comments indicate the various types of directives.

```
# Global directives

archmax = li 8G
examine = noscan
scanlist_squash = on

# limit the drives

drives = stk50 3

# File selection

fs = samfs1
logfile = /var/adm/samfs1.log
archive_archset .
  1 -norelease 5m
  2 -norelease 5m

fs = samfs2
logfile = /var/adm/samfs2.log
archive_archset .
  1 -norelease 5m
  2 -norelease 5m

fs = samfs3
logfile = /var/adm/samfs3.log
archive_archset .
  1 -norelease 5m
  2 -norelease 5m

fs = samfs4
logfile = /var/adm/samfs4.log
archive_archset .
  1 -norelease 5m
  2 -norelease 5m

fs = samfs5
logfile = /var/adm/samfs5.log
archive_archset .
  1 -norelease 5m
  2 -norelease 5m

# The below information is for tape archiving.
# The recycler is not set up to actually recycle. It is set up for
# data checks and file recovery.

params allsets -sort path -offline_copy stageahead -reserve set
allsets -recycle_ignore allsets -recycle_hwm 50 -recycle_mingain 90 -recycle_vsncount 1
allsets.1 -startage 6h -startsize 8G -startcount 90000 -drives 3 -drivemin 10G
allsets.2 -startage 10h -startsize 12G -startcount 90000 -archmax 12G -drives 3 -drivemin 10G
endparams

# Define VSNS for archive sets

vsns
archive.1 li .*
archive.2 li .*
endvsns
```

Example - No archiver.cmd File

This example illustrates the action of the archiver when no `archiver.cmd` file is used in a SAM-QFS environment that has one file system, an optical automated library with two drives, and six cartridges.

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

```
# archiver -lv
Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh
Archive media:
media:lt archmax: 512.0M Volume overflow not selected
media:mo archmax: 4.8M Volume overflow not selected
```

The following `archiver -lv` output indicates that the archiver uses two drives. It lists the 12 volumes, storage capacity, and VSNs with available space.

```
Archive libraries:
Device:hp30 drives_available:2 archive_drives:2
Catalog:
mo.optic00 capacity: 1.2G space: 939.7M -il-o-------
mo.optic01 capacity: 1.2G space: 934.2M -il-o-------
mo.optic02 capacity: 1.2G space: 781.7M -il-o-------
mo.optic03 capacity: 1.2G space: 1.1G -il-o-------
mo.optic10 capacity: 1.2G space: 85.5M -il-o-------
mo.optic11 capacity: 1.2G space: 0 -il-o-------
mo.optic12 capacity: 1.2G space: 618.9k -il-o-------
mo.optic13 capacity: 1.2G space: 981.3M -il-o-------
mo.optic20 capacity: 1.2G space: 1.1G -il-o-------
mo.optic21 capacity: 1.2G space: 1.1G -il-o-------
mo.optic22 capacity: 1.2G space: 244.9k -il-o-------
mo.optic23 capacity: 1.2G space: 1.1G -il-o-------
```

The following `archiver -lv` output shows that the archive set `samfs` includes both metadata and data files. The archiver makes one copy of the files when their archive age reaches the default four minutes (240 seconds).

```
Archive file selections:
Filesystem samfs Logfile:
samfs Metadata
copy:1 arch_age:240
samfs1 path:.
copy:1 arch_age:240
```

The following `archiver -lv` output shows the files in the archive sets archived to the volumes in the indicated order.

```
Archive sets:
allsets
samfs.1
media: mo (by default)
Volumes:
optic00
optic01
optic02
optic03
optic10
optic12
optic13
optic20
optic21
optic22
optic23
Total space available: 8.1G
```

Example - Data Files Archived Separately From Metadata

This example shows how to separate data files into two archive sets separate from the metadata. The environment includes a manually mounted DLT tape drive in addition to an optical automated library. The big files are archived to tape, and the small files are archived to optical cartridges.

The following example shows the content of the `archiver.cmd` file.

```
# archiver -lv -c example2.cmd
Reading archiver command file "example2.cmd"
1: # Example 2 archiver command file
2: # Simple selections based on size
3:
4: logfile = /var/opt/SUNWsamfs/archiver/log
5: interval = 5m
6:
7: # File selections.
8: big . -minsize 500k
9: all .
10: 1 30s
11:
12: vsns
13: samfs.1 mo .*0[0-2] # Metadata to optic00 - optic02
14: all.1 mo .*0[3-9] .*[1-2][0-9] # All others for files
15: big.1 lt .*
16: endvsns
```

The following `archiver -lv` output shows the media and drives to be used.

```
Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh
Archive media:
media:lt archmax: 512.0M Volume overflow not selected
media:mo archmax: 4.8M Volume overflow not selected
Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
Catalog:
mo.optic00 capacity: 1.2G space: 939.7M -il-o-------
mo.optic01 capacity: 1.2G space: 934.2M -il-o-------
mo.optic02 capacity: 1.2G space: 781.7M -il-o-------

mo.optic03 capacity: 1.2G space: 1.1G -il-o-------
mo.optic04 capacity: 1.2G space: 983.2M -il-o-------
mo.optic10 capacity: 1.2G space: 85.5M -il-o-------
mo.optic11 capacity: 1.2G space: 0 -il-o-------
mo.optic12 capacity: 1.2G space: 618.9k -il-o-------
mo.optic13 capacity: 1.2G space: 981.3M -il-o-------
mo.optic20 capacity: 1.2G space: 1.1G -il-o-------
mo.optic21 capacity: 1.2G space: 1.1G -il-o-------
mo.optic22 capacity: 1.2G space: 244.9k -il-o-------
mo.optic23 capacity: 1.2G space: 1.1G -il-o-------
Device:lt40 drives_available:0 archive_drives:0
Catalog:
lt.TAPE01 capacity: 9.5G space: 8.5G -il-o-------
lt.TAPE02 capacity: 9.5G space: 6.2G -il-o-------
lt.TAPE03 capacity: 9.5G space: 3.6G -il-o-------
lt.TAPE04 capacity: 9.5G space: 8.5G -il-o-------
lt.TAPE05 capacity: 9.5G space: 8.5G -il-o-------
lt.TAPE06 capacity: 9.5G space: 7.4G -il-o-------
```

> Note -
> The `archiver(1M) -lv` command shows only VSNs with space available.

The following `archiver -lv` output shows the organization of the file system. Files bigger than 512000 bytes (500 kilobytes) are archived after four minutes. All other files are archived after 30 seconds.

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

The following `archiver -lv` output shows the division of the archive sets among the removable media.

```
Archive sets:
allsets
all.1
media: mo
Volumes:
optic03
optic04
optic10
optic12
optic13
optic20
optic21
optic22
optic23
Total space available: 6.3G
big.1
media: lt
Volumes:
TAPE01
TAPE02
TAPE03
TAPE04
TAPE05
TAPE06
Total space available: 42.8G
samfs.1
media: mo
Volumes:
optic00
optic01
optic02
Total space available: 2.6G
```

Example - User and Data Files Archived to Various Media

In this example, user files and project data files are archived to various media. Files from the directory `data` are segregated by size to optical and tape media. Files assigned to the group ID `pict` are assigned to another set of volumes. Files in the directories `tmp` and `users/bob` are not archived. Archiving is performed at 15-minute intervals, and an archiving record is kept.

```
# archiver -lv -c example3.cmd
Reading archiver command file "example3.cmd"
1: # Example 3 archiver command file
2: # Segregation of users and data
3:
4: interval = 30s
5: logfile = /var/opt/SUNWsamfs/archiver/log
6:
7: no_archive tmp
8:
9: fs = samfs
10: no_archive users/bob
11: prod_big data -minsize 50k
12: 1 1m 30d
13: 2 3m
```

```
14: prod data
15: 1 1m
16: proj_1 projs/proj_1
17: 1 1m
18: 2 1m
19: joe . -user joe
20: 1 1m
21: 2 1m
22: pict . -group pict
23: 1 1m
24: 2 1m
25:
26: params
27: prod_big.1 -drives 2
28: prod_big.2 -drives 2
29: endparams
30:
31: vsns
32: samfs.1 mo optic0[0-1]$
33: joe.1 mo optic01$
34: pict.1 mo optic02$
35: pict.2 mo optic03$
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/opt/SUNWsamfs/scripts/archiver.sh
Archive media:
media:lt archmax: 512.0M Volume overflow not selected
media:mo archmax: 4.8M Volume overflow not selected
Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
Catalog:
mo.optic00 capacity: 1.2G space: 939.7M -il-o-------
mo.optic01 capacity: 1.2G space: 934.2M -il-o-------
mo.optic02 capacity: 1.2G space: 781.7M -il-o-------
mo.optic03 capacity: 1.2G space: 1.1G -il-o-------
mo.optic04 capacity: 1.2G space: 983.2M -il-o-------
mo.optic10 capacity: 1.2G space: 85.5M -il-o-------
mo.optic11 capacity: 1.2G space: 0 -il-o-------
mo.optic12 capacity: 1.2G space: 618.9k -il-o-------
mo.optic13 capacity: 1.2G space: 981.3M -il-o-------
mo.optic20 capacity: 1.2G space: 1.1G -il-o-------
mo.optic21 capacity: 1.2G space: 1.1G -il-o-------
mo.optic22 capacity: 1.2G space: 244.9k -il-o-------
mo.optic23 capacity: 1.2G space: 1.1G -il-o-------
Device:lt40 drives_available:0 archive_drives:0
Catalog:
lt.TAPE01 capacity: 9.5G space: 8.5G -il-o-------
lt.TAPE02 capacity: 9.5G space: 6.2G -il-o-------
lt.TAPE03 capacity: 9.5G space: 3.6G -il-o-------
lt.TAPE04 capacity: 9.5G space: 8.5G -il-o-------
lt.TAPE05 capacity: 9.5G space: 8.5G -il-o-------
lt.TAPE06 capacity: 9.5G space: 7.4G -il-o-------
Archive file selections:
Filesystem samfs Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
copy:1 arch_age:240
no_archive Noarchive path:users/bob
prod_big path:data minsize:50.2k
copy:1 arch_age:60 unarch_age:2592000
copy:2 arch_age:180
prod path:data
copy:1 arch_age:60
proj_1 path:projs/proj_1
copy:1 arch_age:60
copy:2 arch_age:60
joe path:. uid:10006
copy:1 arch_age:60
copy:2 arch_age:60
pict path:. gid:8005
copy:1 arch_age:60
```

```
copy:2 arch_age:60
no_archive Noarchive path:tmp
samfs path:.
copy:1 arch_age:240
Archive sets:
allsets
joe.1
media: mo
Volumes:
optic01
Total space available: 934.2M
joe.2
media: lt
Volumes:
TAPE01
TAPE02
Total space available: 14.7G
pict.1
media: mo
Volumes:
optic02
Total space available: 781.7M
pict.2
media: mo
Volumes:
optic03
Total space available: 1.1G
prod.1
media: mo
Volumes:
optic20
optic21
optic22
optic23
Total space available: 3.3G
prod_big.1
media: lt drives:2
Volumes:
TAPE03
TAPE04
Total space available: 12.1G
prod_big.2
media: lt drives:2
Volumes:
TAPE05
TAPE06
Total space available: 16.0G
proj_1.1
media: mo
Volumes:
optic10
Total space available: 85.5M
proj_1.2
media: mo
Volumes:
optic12
optic13
Total space available: 981.9M
samfs.1
media: mo
Volumes:
optic00
```

```
optic01
Total space available: 1.8G
```

Example - User and Data Files Archived to Optical Media

In this example, user files and project data files are archived to optical media.

Four VSN pools are defined. Three pools are used for user, data, and project, and one is a scratch pool. When `proj_pool` runs out of media, it relies on `scratch_pool` to reserve volumes. This example shows how to reserve volumes for each archive set based on the set component, owner component, and file system component. Archiving is performed at 10-minute intervals, and an archiving log is kept.

The following example shows the `archiver.cmd` file and archiver output.

```
Reading archiver command file "example4.cmd"
1: # Example 4 archiver command file
2: # Using 4 VSN pools
3:
4: interval = 30s
5: logfile = /var/opt/SUNWsamfs/archiver/log
6:
7: fs = samfs
8: users users
9: 1 10m
10:
11: data data
12: 1 10m
13:
14: proj projects
15: 1 10m
16:
17: params
18: users.1 -reserve user
19: data.1 -reserve group
20: proj.1 -reserve dir -reserve fs
21: endparams
22:
23: vsnpools
24: users_pool mo optic0[1-3]$
25: data_pool mo optic1[0-1]$
26: proj_pool mo optic1[2-3]$
27: scratch_pool mo optic2.$
28: endvsnpools
29:
30: vsn
31: samfs.1 mo optic00
32: users.1 mo -pool users_pool -pool scratch_pool
33: data.1 mo -pool data_pool -pool scratch_pool
34: proj.1 mo -pool proj_pool -pool scratch_pool
35: endvsns
Notify file: /etc/opt/SUNWsamfs/scripts/archiver.sh
Archive media:
media:mo archmax: 4.8M Volume overflow not selected
Archive libraries:
Device:hp30 drives_available:0 archive_drives:0
Catalog:
mo.optic00 capacity: 1.2G space: 939.7M -il-o-------
mo.optic01 capacity: 1.2G space: 934.2M -il-o-------
mo.optic02 capacity: 1.2G space: 781.7M -il-o-------
mo.optic03 capacity: 1.2G space: 1.1G -il-o-------
mo.optic04 capacity: 1.2G space: 983.2M -il-o-------
mo.optic10 capacity: 1.2G space: 85.5M -il-o-------
mo.optic11 capacity: 1.2G space: 0 -il-o-------
mo.optic12 capacity: 1.2G space: 618.9k -il-o-------
mo.optic13 capacity: 1.2G space: 981.3M -il-o-------
mo.optic20 capacity: 1.2G space: 1.1G -il-o-------
mo.optic21 capacity: 1.2G space: 1.1G -il-o-------
mo.optic22 capacity: 1.2G space: 244.9k -il-o-------
mo.optic23 capacity: 1.2G space: 1.1G -il-o-------
```

```
Archive file selections:
Filesystem samfs Logfile: /var/opt/SUNWsamfs/archiver/log
samfs Metadata
copy:1 arch_age:240
users path:users
copy:1 arch_age:600
data path:data
copy:1 arch_age:600
proj path:projects
copy:1 arch_age:600
samfs path:.
copy:1 arch_age:240
VSN pools:
data_pool media: mo Volumes:
optic10
Total space available: 85.5M
proj_pool media: mo Volumes:
optic12
optic13
Total space available: 981.9M
scratch_pool media: mo Volumes:
optic20
optic21
optic22
optic23
Total space available: 3.3G
users_pool media: mo Volumes:
optic01
optic02
optic03
Total space available: 2.7G
Archive sets:
allsets
data.1
reserve:/group/
media: mo
Volumes:
optic10
optic20
optic21
optic22
optic23
Total space available: 3.4G
proj.1
reserve:/dir/fs
media: mo
Volumes:
optic12
optic13
optic20
optic21
optic22
optic23
Total space available: 4.2G
samfs.1
media: mo
Volumes:
optic00
Total space available: 939.7M
users.1
reserve:/user/
media: mo
Volumes:
optic01
optic02
optic03
optic20
optic21
optic22
```

```
optic23
Total space available: 6.0G
```

About Disk Archiving

Archiving is the process of copying a file from online disk to archive media. With disk archiving, the archive medium is online disks in a file system.

Disk archiving can be implemented so that the files from one file system are archived to another file system on the same host or to another file system on a different host. When disk archiving is configured for two host systems, the systems act as a client and a server. The host system where the source files reside is the client and the host system where the archive copies are written is the server.

The file system to which the archive files are written can be any UNIX file system. However, if disk archive copies are written to a different host, that host must have at least one QFS or SAM-QFS file system installed on it.

The archiver treats files archived to disk volumes in the same way as files archived to volumes in a library. You can make one, two, three, or four archive copies. If you are making multiple archive copies, you can write one of the archive copies to disk volumes and write the other archive copies to removable media volumes. In addition, if you archive to disk volumes that are in an archived file system, the archive copies are also archived according to the `archiver.cmd` file rules for that file system.

When you plan disk archiving for your site, consider the following guidelines:

- Create disk volumes on a different disk from the one on which the original files reside.
- Make more than one archive copy and write to more than one type of archive media. For example, create archive copy 1 to disk volumes, copy 2 to tape, and copy 3 to magneto-optical disk.
- If you are archiving files to a file system on a server system, the archive copies can also be written to removable media cartridges in a library attached to that server system.

The following list summarizes some of the similarities and differences between archiving to disk volumes and archiving to removable media:

- Archive copies in disk volumes are not recorded in a catalog.
- Archive copies in disk volumes do not appear in the historian.
- Archiving to disk volumes does not rely on entries in the `mcf` file. Instead, you specify disk archive sets in the `archiver.cmd` file and you define disk volumes in `/etc/opt/SUNWsamfs/diskvols.conf`.
- To archive to disk volumes, you must define disk archive sets in the `archiver.cmd` file before mounting the file system. When you archive to removable media volumes, you can begin archiving after the file system is mounted without changing any of the default values in the `archiver.cmd` file.

About the diskvols.conf File

Disk archiving does not rely on entries in the `mcf` file. You specify disk archive sets in the `archiver.cmd` file and you define disk volumes in `/etc/opt/SUNWsamfs/diskvols.conf`.

Create the `diskvols.conf` file on the system where the source files reside. Depending on where the archive copies are written, the `diskvols.conf` file contains the following information:

- If the archive copies are written to a file system on the same host system, the `diskvols.conf` file defines the VSNs and the paths to each VSN.
- If the archive copies are written to a different host system, the `diskvols.conf` file contains the host name of that server system. In this case, create another `diskvols.conf` file on the server system that specifies the host names of the client systems.

> Caution
> Be very careful when configuring the recycler if you are using disk archiving in an environment with multiple SAM-QFS servers. The `diskvols.conf` file for each SAM-QFS server must point to a unique set of disk volume resource specifications (disk archiving target directories). If any of the target directories are shared by the SAM-QFS servers, running the recycler from one SAM-QFS server destroys the disk archive data that is managed by the other SAM-QFS server.

The `diskvols.conf` file must contain the following information:

```
<VSN_name> [<host_name>:] <path>
```

Field Name	Content
VSN_name	A unique alphanumeric name of up to 31 characters for the disk VSN to receive the archive copies.
host_name	The name of the host, followed by a colon character (:), to which archive copies are written. If you are archiving to a disk on another host, specify the name of the destination server. If you are archiving to a file system that resides on the same server as the source file system, do not specify the host name.
path	The full path, relative to the mount point, to the directory that receives the archive files. This directory must be in place before archiving can start, and the destination file system must be mounted. For example, if archive copies are written to the vsns directory in the archivefs1 file system, specify /archivefs1/vsns in the path field. Create the destination directory with write permission granted only to root.

The following additional rules apply to the diskvols.conf file:

- Start each comment line with a pound character (#). All text to the right of the # is ignored.
- To continue a line, put an apostrophe character (') at the end.

For more information, see the diskvols.conf(4) man page.

Disk Archiving Directives

When archiving to disk volumes, the archiver recognizes the directives that define archive sets and recycling and ignores directives that pertain to removable media cartridges. The archiver recognizes the directives in the following sections:

- Archive Set Copy Parameters
- Archive Directives
- Recycling Directives
- vsn Directives
- clients and endclients Directives
- -recycle minobs Recycler Directive

Archive Set Copy Parameters

All the parameters described in Archive Set Copy Parameters are valid except for the following:

- -reserve method
- -tapenonstop

To configure an archive set to write multiple, simultaneous disk archive streams, use the -drives parameter. In this configuration, volumes are selected in a round robin manner starting with the volume that has the highest percentage of available space. However, if the parameter '-fillvsns' is specified, the volume with the least percentage
remaining space is selected first.

Archive Directives

All the directives described in Archive Directives are valid except for the following:

- ovflmin min-size

Recycling Directives

All of the directives described in recycling directives are valid except for the following:

- -recycle_dataquantity size
- -recycle_vsncount count
- recycle_hwm

vsn Directives

The following directives are valid:

- vsns and endvsns
- vsnpools and endvsnpools

Disk volumes are supported in the VSN associations section and are defined with a dk media type. The volumes are identified by one or more VSN expression keywords. You can also specify VSN pools from which disk volumes are to be selected as shown in the following example.

```
vsnpools
data_pool dk disk0[0-5]
endvsnpools

vsns
arset0.1 dk disk10 disk1[2-5]
arset1.1 dk -pool data_pool
endvsns
```

Disk archiving can be carried out on the Sun StorageTek 5800 system. The Sun StorageTek 5800 is an online storage appliance with an integrated hardware and software architecture in which the disk-based storage nodes are arranged in a symmetric cluster. The media abbreviation for Sun StorageTek 5800 disk archives in the `vsns` directives is `cb`.

> Note -
> If you are using the disk volumes on the Sun StorageTek 5800 for archiving, be aware that the Sun StorageTek 5800 is not a traditional file system and the security considerations are different from other types of disk storage. Read the Sun StorageTek 5800 documentation for more information.

clients and endclients Directives

The `clients` and `endclients` directives are valid. If you archive files from a client host to a server host, the server system must have a `diskvols.conf` file that contains the name of the client system. The format for these directives is shown in following example. For client-system, specify the host name of the client system that contains the source files.

```
clients
_client-system1_

_client-system2_
...
endclients
```

-recycle minobs Recycler Directive

The `-recycle_minobs` *`percent`* recycler directive is valid. This option is used to set a threshold for the recycler's rearchiving process for disk archives. The default threshold is 50 percent. When the percentage of obsolete files within an archived tar file on the disk reaches this threshold, the recycler moves the valid files from the archive into a new tar file. When all of the valid files have been moved, the original tar file is marked as a candidate for removal from the disk archive. This option is ignored for removable media recycling.

How to Enable Disk Archiving on the Host That Contains the Files to Be Archived

Perform this procedure on the host system that contains the files to be archived. As an alternative, you can use the SAM-QFS Manager interface to specify an archive policy that archives to disk volumes. This action updates both the `archiver.cmd` file and the `diskvols.conf` file.

> Note -
> If you are configuring a Sun QFS file system for the first time at your site and have therefore not yet installed the SAM-QFS software on another host, you must write the archive copies to disk volumes in a file system that is on the same host as the source files. If you configure a QFS file system on another host at a later time, you can modify your configuration files accordingly.

1. Become superuser on the host system that contains the files you want to archive.
2. Create or open the file `/etc/opt/SUNWsamfs/archiver.cmd`.
3. Add disk archive set directives as in the following example:

```
#
vsns
archset1.1 dk disk01
archset2.1 dk disk02
archset3.1 dk disk03
endvsns
```

Disk archiving can also be carried out on the Sun StorageTek 5800 system. The Sun StorageTek 5800 is an online storage appliance with an integrated hardware and software architecture in which the disk-based storage nodes are arranged in a symmetric cluster. The media abbreviation for Sun StorageTek 5800 disk archives in the `vsns` directives is `cb`.
For more information about specifying archive sets, see the `archiver.cmd`(4) man page or see the Archive Set Directives (archiver.cmd) page.

4. Save and close the `archiver.cmd` file.
5. Create a file named `diskvols.conf`.
6. Specify the directories to which the archive copies will be written.
The following example shows a `diskvols.conf` file that archives files from three archive sets. The disk volumes named `disk01` and `disk02` reside in a file system on the server system named `otherserver`. Disk volume `disk03` resides on the same host as the files to be archived.

```
# This is file sourceserver:/etc/opt/SUNWsamfs/diskvols.conf
# on the client
#
# VSN_name [host_name:] path
#
disk01 otherserver:/sam/archset1
disk02 otherserver:/sam/archset2
disk03 /sam/archset3
```

7. Save and close the `diskvols.conf` file.
8. Create directories in the file system to which the archive copies will be written. For example:

```
# mkdir sam
# cd sam
# mkdir archset1
# mkdir archset2
```

9. Verify the syntax in the `archiver.cmd` file:

```
# archiver -lv
```

10. If any errors are found, correct them before proceeding.

How to Configure Disk Archiving on the Host to Which the Archive Copies Will Be Written

Perform this procedure only if you are writing your archive copies to a host system that is different from the host system upon which the source files reside. At least one QFS or SAM-QFS file system must be created on this host. If you create source files and write archive copies to the same host system, you do not need to perform this procedure.

> Note -
> You can use the SAM-QFS Manager interface to enable disk archiving by specifying an archive policy that archives to disk VSNs. This action updates both the `archiver.cmd` file and the `diskvols.conf` file.

In this situation, you are creating a client/server environment:

- The client is the host that contains the source files.
- The server is the host to which the archive copies are written.

1. Become superuser on the server.

2. Create or open the file `/etc/opt/SUNWsamfs/archiver.cmd`.
3. Edit the `archiver.cmd` file to add disk archive set directives as in the following example:

```
#
vsns
archset1.1 dk disk01
archset2.1 dk disk02
archset3.1 dk disk03
endvsns
```

   Disk archiving can also be carried out on the Sun StorageTek 5800 system. The Sun StorageTek 5800 is an online storage appliance with an integrated hardware and software architecture in which the disk-based storage nodes are arranged in a symmetric cluster. The media abbreviation for Sun StorageTek 5800 disk archives in the `vsns` directives is `cb`.
   For more information about specifying archive sets, see the `archiver.cmd`(4) man page or see the Archive Set Directives (archiver.cmd) page.

4. Save and close the file.
5. Change to the file system to which you want to write the archive copies. For example:

```
# cd /ufs1
```

6. Create directories in the file system. For example:

```
# mkdir sam
# cd sam
# mkdir archset1
# mkdir archset2
```

7. Create the file `/etc/opt/SUNWsamfs/diskvols.conf`.
8. Specfiy the `clients` and `endclients` directives and the name of the client. The name of the client in the following example is `sourceserver`.

```
# This is
# file destination_server:/etc/opt/SUNWsamfs/diskvols.conf
# on the server
#
clients
sourceserver
endclients
```

9. Save and close the `diskvols.conf` file.

How to Enable Disk Archiving

You can enable disk archiving at any time. The procedure assumes you are adding disk archiving to an existing archiving configuration.

1. Make certain that the host to which you want to write your disk archive copies has at least one QFS or SAM-QFS file system installed on it.
2. Become superuser on the host system that contains the files to be archived.
3. Follow the How to Enable Disk Archiving on the Host That Contains the Files to Be Archived procedure or How to Configure Disk Archiving on the Host to Which the Archive Copies Will Be Written procedure.
4. On the host that contains the files to be archived, use the following command to propagate the configuration file changes and restart the system.

```
# samd config
```

5. If you are archiving to disk on a different host, follow these steps:
   a. Become superuser on the host system to which the archive copies are written.
   b. Use the `samd config` command to propagate the configuration file changes and restart the destination system.

6. If you are archiving to a Sun StorageTek 5800 system, you must upgrade the Sun StorageTek 5800 metadata schema configuration. Follow the procedures documented in the Sun StorageTek 5800 System Administration Guide and use the XML overlay in the following example to define the metadata that is used by SAM-QFS.

Example - Metadata Schema for SAM-QFS on an STK 5800

```
<<?xml version="1.0" encoding="UTF-8"?>

<metadataConfig>
<schema>
<namespace name="com">
<namespace name="sun">
<namespace name="samfs">
<field name="archiveId" type="string" indexable="true"/>
<field name="fileName" type="string" indexable="true"/>
<field name="modTime" type="time" indexable="true"/>
</namespace>
</namespace>
</namespace>
</schema>

<fsViews>
</fsViews>

</metadataConfig>
```

Disk Archiving Examples

The following examples show disk archiving configurations.

Example 1

In this example, VSNs identified as `disk01`, `disk02`, and `disk04` are written to `pluto`, the host system where the original source files reside. VSN `disk03` is written to a VSN on server system `mars`.

The following example shows the `diskvols.conf` file that resides on client system `pluto`.

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on pluto
# VSN Name [Host Name:]Path
#
disk01 /sam_arch1
disk02 /sam_arch2/proj_1
disk03 mars:/sam_arch3/proj_3
disk04 /sam_arch4/proj_4
```

The following example shows the `diskvols.conf` file on server system `mars`.

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on mars
#
clients
pluto
endclients
```

The following example shows a fragment of the `archiver.cmd` file on `pluto`.

```
vsns
arset1.2 dk disk01
arset2.2 dk disk02 disk04
arset3.2 dk disk03
endvsns
```

Example 2

In this example, file /sam1/testdir0/filea is in the archive set for arset0.1, and the archiver copies the content of this file to the destination path /sam_arch1.
The following example shows the diskvols.conf file.

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf
#
# VSN Name [Host Name:]Path
#
disk01 /sam_arch1
disk02 /sam_arch12/proj_1
```

The following example shows the archiver.cmd file lines that pertain to disk archiving:

```
.
vsns
arset0.1 dk disk01
endvsns .
```

The following example shows output from the sls(1) command for file filea, which was archived to disk. Note the following for copy 1:

- dk is the media type for disk archive media
- disk01 is the VSN
- f192 is the path to the disk archive tar(1) file

```
# sls -D /sam1/testdir0/filea
/sam1/testdir0/filea:
mode: -rw-r----- links: 1 owner: root group: other
length: 797904 admin id: 0 inode: 3134.49
archdone;
copy 1: ---- Dec 16 14:03 c0.1354 dk disk01 f192
access: Dec 19 10:29 modification: Dec 16 13:56
changed: Dec 16 13:56 attributes: Dec 19 10:29
creation: Dec 16 13:56 residence: Dec 19 10:32
```

Example 3

In this example, file /sam2/my_proj/fileb is on client host snickers in archive set arset0.1, and the archiver copies the content of this file to the destination path /sam_arch1 on server host mars.

The following example shows the diskvols.conf file on snickers.

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on snickers
#
# VSN Name [Host Name:]Path
#
disk01 mars:/sam_arch1
```

The following example shows the diskvols.conf file on mars.

```
# This is file /etc/opt/SUNWsamfs/diskvols.conf on mars
#
clients
snickers
endclients
```

The following example shows the directives in the archiver.cmd file that relate to this example.

```
.
vsns
arset0.1 dk disk01
endvsns .
```

Configuring the Recycler

Contents

Configuring the Recycler

This page describes the recycling process and directives.

Recycling Methods

You initiate recycling by entering the `sam-recycler`(1M) command either manually or through a `cron`(1) job. The following table shows the recycling methods.

Recycling Methods and Media Types

Recycling Method	Archive Media	Where To Configure
By automated library	Removable media cartridges	`recycler.cmd` `recycler.sh`

By archive set	Removable media cartridges	`recycler.cmd` (optional, for VSNs not covered in archive set)
		`recycler.sh`
		`archiver.cmd`

By archive set	Disks	`archiver.cmd`

Before configuring the recycler, note the following:

- Directives in the `archiver.cmd` file control recycling by archive set.
  Directives in the `recycler.cmd` file control recycling by library. In addition, the `recycler.cmd` file controls general recycler behavior. For information about recycler directives, see Creating a `recycler.cmd` File.
- Do not recycle volumes that contain removable media files. You create removable media files by using the `request`(1) command. A volume with removable media files can never be drained.
- Do not run the recycler while performing maintenance on a file system. The recycler uses the `.inodes` file and the `mcf` file to determine whether files are current or expired and to identify the devices associated with a file system. Absence of proper information in the `.inodes` and `mcf` files can cause current archived data to appear as expired and be recycled.
- All file systems must be mounted when the recycler is run. If you are recycling from online disk, the file system that contains the disk volumes must be mounted and the host system must be accessible.

Caution
Take extreme care when configuring the recycler if you are using disk archiving in an environment with multiple SAM-QFS servers. The `diskvols.conf` file for each SAM-QFS server must point to a unique set of disk archiving target directories. If any of these directories are shared by different SAM-QFS servers, running the recycler from one SAM-QFS server will destroy the disk archive data that is managed by the other SAM-QFS server.

Controlling Recycling

You can enable and disable the recycle process using parameters in the command files.
When you are configuring the recycler and testing the results, edit the appropriate file to add its parameter.

If You Recycle By...	Add The Parameter
Archive Set	`-recycle_ignore` parameter to the `archiver.cmd` file.
Automated Library	`ignore` parameter to the `recycler.cmd` files.

When you are ready to use the recycler, edit the files to remove the parameter.
As an alternative, you can use the SAM-QFS Manager. For more information, see the SAM-QFS Manager online help.

How to Start the Recycler

Follow these instructions to run the recycler.

1. Run the `sam-recycler`(1M) command.

```
# sam-recycler
```

   The recycler reads the `recycler.cmd` file.

2. Examine the standard output log, SAM-QFS log, and `/var/adm/messages` for any error messages from the recycler.
   Correct your files if errors appear.

When the recycler is initiated, the default recycler settings specified in library Directive: Specifying Recycling for an Entire Automated Library take effect. For more information on the recycler, see the `sam-recycler`(1M) man page.

How to Run the Recycler Periodically

If the system is performing in a routine manner, you can make a `crontab` entry. The frequency you choose depends on your site's conditions.
For instructions about creating a `crontab` entry, see the `cron`(1M) man page.

The following example entry in root's `crontab` file specifies that the `cron` daemon run the recycler every five minutes after every odd-numbered hour:

```
5 1,3,5,7,9,11,13,15,17,19,21,23 * * * /opt/SUNWsamfs/sbin/sam-recycler
```

Configuring Recycling on Removable Media Cartridges

If you are recycling archive copies on cartridges in a library, create a `recycler.cmd` file.
If you are recycling by archive set, configure each library in the `recycler.cmd` file. This ensures that VSNs that do not fall into an archive set and can be recycled if needed.
Create a `recycler.sh` file to complete the operation.

Creating a **`recycler.cmd`** File

The `recycler.cmd` file contains general recycling directives. It can also contain directives for each library in the SAM-QFS environment. A typical `recycler.cmd` file contains the following directive lines:

- A `logfile=` directive to specify a recycler log file.
- One or more directives for each library that contains volumes to be recycled. This line must contain the family set name (from the `mcf` file) for the library being recycled. The family set name identifies the library to the recycler. For information, see Using Recycling Directives.
- During testing, include the `ignore` keyword. You remove the `ignore` keyword in a later step in this process.

1. Become superuser.
2. In the `/etc/opt/SUNWsamfs/recycler.cmd` file, add one or more directives to control recycler activity.
3. Save and close the file.
   Aa an alternative, you can create a `recycler.cmd` file using SAM-QFS Manager software. For more information, see the SAM-QFS Manager online help.

Example - Sample **`recycler.cmd`** File

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

The `recycler.cmd` file accepts the directives described in the following sections:

- logfile Directive: Specifying a Log File
- no_recycle Directive: Preventing Recycling
- library Directive: Specifying Recycling for an Entire Automated Library

logfile Directive: Specifying a Log File

The `logfile` directive creates a recycler log file. This directive has the following format:

```
logfile = <filename>
```

For filename, specify the path to the log file.
The following is an example of a `logfile=` directive line:

```
logfile=/var/adm/recycler.log
```

Example - Sample Recycler Log File for Removable Media Cartridges

The following example shows a sample recycler log file for recycling removable media cartridges.

```
========== Recycler begins at Wed Dec 12 14:05:21 2001 ===========
Initial 2 catalogs:
0 Family: m160 Path: /var/opt/SUNWsamfs/catalog/m160
Vendor: ADIC Product: Scalar 100
SLOT ty capacity space vsn
0 at 25.0G 25.0G CLN005
1 at 48.5G 6.1G 000003
2 at 48.5G 32.1G 000004
3 at 48.5G 35.1G 000005
4 at 48.5G 44.6G 000044
5 at 48.5G 45.1G 000002
6 at 48.5G 45.9G 000033
7 at 48.5G 48.5G 000001
Total Capacity: 364.8G bytes, Total Space Available: 282.3G bytes
Volume utilization 22%, high 95% VSN_min 50%
Recycling is ignored on this robot.
1 Family: hy Path: /var/opt/SUNWsamfs/catalog/historian
Vendor: Sun SAM-FS Product: Historian
SLOT ty capacity space vsn
(no VSNs in this media changer)
Total Capacity: 0 bytes, Total Space Available: 0 bytes
Volume utilization 0%, high 95% VSN_min 50%
Recycling is ignored on this robot.
8 VSNs:
---Archives--- -----Percent----- m160
----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 87 13 m160:at:000003
no-data VSN 0 0 0 33 67 m160:at:000004
no-data VSN 0 0 0 27 73 m160:at:000005
no-data VSN 0 0 0 8 92 m160:at:000044
no-data VSN 0 0 0 7 93 m160:at:000002
no-data VSN 0 0 0 5 95 m160:at:000033
empty VSN 0 0 0 0 100 m160:at:CLN005
empty VSN 0 0 0 0 100 m160:at:000001
Recycler finished.
========== Recycler ends at Wed Dec 12 14:05:32 2001 ===========
```

no_recycle Directive: Preventing Recycling

The `no_recycle` directive disables recycling of volumes. This directive has the following format:

```
no_recycle <media-type> <VSN-regexP> [<VSN-regexP>...]
```

Argument	Meaning
media-type	A media type from the `mcf`(4) man page. You can disable recycling of volumes stored on particular type of media
VSN-regexp	One or more space-separated regular expressions to describe the volumes. You can disable recycling for specific cartridges. For information, see the `regexp`(5) man page or see File Name search-criterion Using Pattern Matching: -name regex.

Example - Preventing Recycling of Specific Cartridges

The following example excludes any tape volumes whose VSN identifiers begin with `DLT`:

```
no_recycle lt DLT.*
```

library Directive: Specifying Recycling for an Automated Library

The library directive enables you to specify various recycling parameters for the VSNs associated with a specific library. This directive has the following format:

```
<library> <parameter> [<parameter>...]
```

For library, specify the library's name as specified in the family set field of the mcf file.
For parameter, specify one or more space-separated parameter keywords from the following table.

Parameter Value	Meaning
-dataquantity size	Maximum amount of data that the recycler can schedule for rearchiving in its efforts to clear volumes of useful data. Default is 1 gigabyte.
-hwm percent	Library high-water mark. Default is 95.
-ignore	Directive that prevents volumes in this library from being recycled. This directive is useful during testing of the recycler.cmd file.
-mail email-address	Email addresses to which recycling email messages are to be sent. By default, no email is sent.
-mingain value	Minimum VSN gain. Default depends on media: • For volumes with less than 200 GByte capacity, the default mingain is 60%. • For volumes with 200 GByte or larger capacity, the default mingain is 90%.
-vsncount count	Maximum number of recycled volumes to be counted. Default is 1.

Example - library Directive

The following example specifies the following for library gr47:

- The library qualifies for recycling when the volumes in the library are 85 percent full.
- The minimum percent gain is 40 percent.
- Only one volume is to be recycled. This is also a default setting.
- Recycling messages are emailed to root.
- No more than 1 gigabyte is to be re-archived. This is the default, so it is not specified in the recycler.cmd file.

```
gr47 -hwm 85 -ignore -mail root -mingain 40
```

The following sections describe the parameters.

-hwm Parameter

By specifying a high-water mark, you set the percentage of media usage below which recycling cannot occur. This percentage is the ratio of the used space in the library to its total capacity. For example, a library that holds ten 20-gigabyte tapes, three of them 100 percent full and the remaining seven each 30 percent full, has the following media utilization percentage:

```
((3* 1.00 + 7 * 0.30) * 20G ) / ( 10 * 20G ) * 100%= 51%
```

This calculation does not distinguish between current data and expired data. It only addresses the amount of media used.

In the example, when the utilization percentage is 51 percent or less, the recycler does not automatically select any of the automated library's VSNs for recycling.

You can force a VSN to be recycled by using the following command to set the recycling flag:

```
# chmed +c lt. <VSN>
```

When the +c flag is set, the archiver does not write any more archive images to the volume. The +c flag can be viewed through the samu(1M) utility. For more information, see the chmed(1M) and samu(1M) man pages. For information about using the samu(1M) operator utility, see the

Sun QFS File System Configuration and Administration Guide.

-mingain Parameter

The minimum VSN gain percentage sets a lower limit on the amount of space to be gained by recycling a cartridge. For example, if a cartridge in an automated library is 95 percent current data and 5 percent is efficient. Setting the minimum gain to 6 percent or more inhibits the recycler from automatically selecting this VSN.

-ignore Parameter

The `-ignore` parameter disables the recycler for a particular library. Use it while you are configuring and testing the recycler.

-mail Parameter

The `-mail` parameter specifies that the recycler sends email messages when recycling occurs on a library. The email message has the following subject line:

```
Robot <robot-name> recycle
```

Example - Sample Recycling Messages

```
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.
```

Creating a **recycler.sh** File

If you are archiving on removable media cartridges, create a `recycler.sh`.
If you are archiving only to disk, do not perform this step.

Determine your site's policy for recycled cartridges. Some sites relabel and reuse the cartridges and some sites remove the cartridges from the automated library to use later for accessing historical files.

The recycler executes the `recycler.sh` script when all the current images from a VSN have been re-archived to another VSN.
The recycler calls the `/opt/SUNWsamfs/scripts/recycler.sh` script with the following arguments:

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

For examples of the script, see the `recycler.sh(1M)` man page or view the `/opt/SUNWsamfs/examples/recycler.sh` script. The latter shows how to relabel a recycled VSN and send mail to the superuser.

Configuring Recycling for Disk Archive Volumes

If you are archiving to disk, you must edit the `archiver.cmd` file to recycle.
If you are recycling by archive set, you must add archive set recycling directives between the `params` and `endparams` directives.
If you are recycling by library, this step is optional.

Editing the **archiver.cmd** File

To edit the `archiver.cmd` file, follow the steps described in About the `archiver.cmd` File.
As an alternative, you can edit the `archiver.cmd` file by using the File System Manager. For more information, see the File System Manager online help.

The following table shows the archive set recycling directives that you can use.

Table - Archive Set Recycling Directives

Directive	Function

`-recycle_dataquantity` size	Limits the amount of data the recycler schedules for re-archiving to clear a disk volume of useful data. By default, a limit is ignored for disk archive recycling.
`-recycle_ignore`	Prevents the archive set from being recycled.
`-recycle_mailaddr` mail-address	Sends recycler messages to the specified email address.
`-recycle_mingain` percent	Limits recycling of volumes in the archive set by setting the `mingain` mark for a disk volume. The `mingain` is expressed as a percentage of the expired data associated with the volume. When the expired data of the volume exceeds the `mingain` percentage, the recycler begins to recycle the volume. The default is 50%.
`-recycle_minobs` percent	Limits the recycler's selection of tar files in volume by setting a threshold for the recycler's rearchiving process of disk archive volumes. When the percentage of expired files within an archived tar file on the disk reaches this threshold, the recycler begins moving the current files from the archive into a new tar file. Once all the current files have been moved, the original tar file is marked as a candidate to be removed from the disk archive. The default is 50%.
`-rearch_stage_copy` copy-number	Sets staging for re-archiving to take place from selected (faster) copies.

For more information about archiver directives, see Configuring the Archiver or the `archiver.cmd`(4) man page.

Recycler Logging for Disk Archives

Example - Recycler Log File for Disk Archive Files

```
---Archives---   -----Percent-----
 ----Status-----    Count    Bytes   Use Obsolete Free   Library:Type:VSN
new candidate          0      0        0    41     59  <none>:dk:disk01
677 files recycled from VSN disk01 (mars:/sam4/copy1)
0 directories recycled from VSN disk01 (mars:/sam4/copy1)
```

Recycling for Archive Copy Retention

As an alternative to the normal recycling process, you can use the `sam-nrecycler`(1M) tool to work with the File System Manager's backup and recovery point features. This tool removes expired archive copies and frees archive volumes to aid in the ability to use SAM-QFS dump files for archive retention. To take advantage of this functionality, you must use this recycler in place of the existing `sam-recycler` command.

The `sam-nrecycler`(1M) tool scans file system metadata and SAM-QFS dump files to determine which volumes contain archive images. You can invoke the tool through the `crontab`(1) file at an off-peak time, or at any time using the `sam-nrecycler` command. The nrecycler identifies all archive images on a removable media volume or in a disk archive tar file by scanning all file system `.inodes` files and specified SAM-QFS dump files. The `nrecycler` can then determine whether volumes contain any archive images. The space on these volumes can be reclaimed. If a removable media volume does not contain any archive images, relabel the cartridge. If a disk archive tar file does not contain any archive images, remove the tar file from the disk archive directory.

When `sam-nrecycler`(1M) detects that a removable media volume contains only free or expired space and is safe to relabel, it invokes the `sam-nrecycler.sh` script. The script can relabel the cartridge using either the original VSN or a new VSN. It can then export the cartridge from the library, or it can perform another user-defined action.
When `sam-nrecycler` detects that a disk archive volume contains only free or expired space, it unlinks the unused disk archive tar file.

You control the actions of the `sam-nrecycler`(1M) toll by including directives in the `/etc/opt/SUNWsamfs/nrecycler.cmd` file. You must also specify the path to the directories that contain the SAM-QFS dump files. The list of directories must be complete and all SAM-QFS dump files must be contained in the directory list.

You can also include a `logfile=` directive line in the `nrecycler.cmd` file to specify an nrecycler log file. The system writes recycling messages and recycling reports to this file.

For more information about `sam-nrecycler`(1M), see the `sam-nrecycler`(1M) man page.

Configuring the Releaser

Contents

Configuring the Releaser

The /etc/opt/SUNWsamfs/releaser.cmd file consists of directive lines that specify site-specific releasing actions. The releaser.cmd file can contain directives for setting the release priority, specifying a log file, and other actions.

For more information about these directives, see the releaser.cmd(4) man page.

Some global releasing directives can be configured using the SAM-QFS Manager software. See the SAM-QFS Manager online help for more information.

How to Run the Releaser Manually

To run the releaser manually, identify the mount point of the file system and the low-water mark that you want the releaser to reach. If the site has a releaser.cmd file that includes this information, use the following command:

```
# /opt/SUNWsamfs/sbin/sam-releaser
```

If the releaser.cmd file does not exist, specify the following arguments.

```
# /opt/SUNWsamfs/sbin/sam-releaser <file-system> <low-water-mark> <weight-size>
```

The command-line arguments override equivalent directives in the releaser.cmd file.

For example, to release files in the /sam1 file system until it reaches 47 percent full, log in as root and type the following:

```
# /opt/SUNWsamfs/sbin/sam-releaser /sam1 47 1.0
```

As the releaser runs, it writes information to your screen and to the releaser log file, if one is specified in the releaser.cmd file. For more information, see the sam-releaser(1M) man page.

How to Specify Release Attributes for All Files in an Archive Set

Most directives in the archiver.cmd file affect archiving, but the archive set assignment directive enables you to specify release attributes that apply to all files in an archive set.

1. Open the archiver.cmd file and locate the archive set assignment directive.
2. Add the -release file attribute to the archive set assignment directive.

```
-release <attribute>
```

Attribute	Meaning

a	Specifies that the files in the archive set are released after the first archive copy is made. Do not use this option if you are making more than one archive copy of each file because copy 1 is staged to make copy 2.
d	Reset to default.
n	Specifies that the files in the archive set are never released.
p	Specifies that the files in the archive set are partially released after archiving.

The attributes for the `-release` directive follow the same conventions as the `release`(1) command.

Planning Releaser Operations

Determine the characteristics of files in cache for your site because loading a tape for only a few kilobytes of data is inefficient. Consider configuring your system to retain small files in cache as in the following example.

Example - Release the Largest Files First

The following example shows the directives to use in the `releaser.cmd` file to release the largest files first.

```
weight_size = 1.0
weight_age = 0.0
```

It is a good practice to retain recently-modified files in cache because it is probable that a recently-modified file is modified again soon. This practice avoids having to stage the file again. In this case, use the second set of age weights (the age since last modified).

Example - Release Oldest Modified Files First

The following example shows the directives to use in the `releaser.cmd` file to weight files from the oldest modified to the most recently modified.

```
weight_size = 0.0
weight_age_access = 0.0
weight_age_modify = 1.0
weight_age_residence = 0.0
```

The previous examples show simple situations. Most sites require refinement of the directives.

Assume that you want to release the largest files first. Your site has hundreds of small files of the same size and several large files. Eventually, the releaser has released all the large files. Then, the releaser starts releasing the small files in random order because they are all the same size and have the same release priority. To prevent this from happening, use the `weight_age` directive. Set `weight_age = 0.01` to release the oldest of the small files first. Or, set `weight_size = 1.0` and `weight_age = 0.01`. These directives violate the largest-first policy by counting smaller, less recently accessed files as better candidates than larger, more recently accessed files. However, you can reduce this effect by making `weight_age` smaller than `weight_size`.

For example, with the previous settings, a 4-kilobyte file that was staged 100 minutes ago and an 8-kilobyte file that was just staged both have the same release priority However, if you set `weight_age` = 0.001, a 4-kilobyte file must have been staged 1,000 minutes ago to have the same priority as the 8-kilobyte file that was just staged.

Testing the Releaser

For assistance in adjusting priority weights, use the `no_release` and `display_all_candidates` directives and then run the releaser manually to obtain a list of candidates in priority order. This method validates your specifications and tests the releaser's actions. You can modify the attributes until you are satisfied with the releaser's actions, and then you remove the `no_release` and `display_all_candidates` directives to implement your configuration.

fs Directive: Specifying Directives for Individual File Systems

You can use the `fs` = family-set-name directive to control the releaser's actions for the named file system.

Directives preceding the first `fs` = directive are global and apply to all file systems. Directives following the `fs` = directive override global directives. The directives described in this page can be used either as global directives or as directives specific to one file system.

This directive has the following format:

```
fs = <family-set-name>
```

For family-set-name, specify the name of a family set in the `mcf` file.
The `releaser.cmd`(4) man page includes examples of the `fs =` directive.

no_release and display_all_candidates Directives: Debugging the Releaser

Use the `no_release` and `display_all_candidates` directives for tuning or debugging the releaser. These directives are helpful during debugging because the releaser writes the names of release candidates to the log file, but it does not physically release them from the file system.

- The `no_release` directive prevents files from being removed from online disk cache. Use this directive to disable the releaser. This directive has the following format:

  ```
  no_release
  ```

- The `display_all_candidates` directive writes the names of all release candidates to the log file. This directive has the following format:

  ```
  display_all_candidates
  ```

list_size Directive: Adjusting the Size of the Releaser Candidate List

Use the `list_size` directive to specify the number of releaser candidates. If you notice that the releaser makes several file system scans before it releases the number of files needed to reach the low-water mark, consider raising this value to a level greater than the default of 10,000. This situation might occur in a file system that contains many small files. You can get information about releaser activities from the releaser log file. This directive has the following format:

```
list_size = <number>
```

For number, specify an integer from 10 through 2,147,483,648.

logfile Directive: Specifying a Log File

When a `logfile` directive is specified in the `releaser.cmd` file, the releaser either appends its activity log to the indicated file, or creates the file if it does not exist. This directive has the following format:

```
logfile = <filename>
```

For filename, specify the name of a log file.

Example - Releaser Log File

```
Releaser begins at Wed Apr 28 17:29:06 2006
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-amld? 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 2006) 10001 min, 500 blks /sam1/testdir0/filevp
10500 (R: Wed Apr 21 18:48:10 CDT 2006) 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 2006
```

The `releaser`(1M) man page describes the information contained in the log file. Because the size of the log increases with each releaser run, plan to rotate the log file, or omit the `logfile` keyword.

min_residence_age Directive: Specifying a Minimum Residence Time

The `min_residence_age` directive enables you to specify the minimum amount of time that a file must reside in a file system before it becomes a candidate for release. This directive has the following format:

```
min_residence_age = <time>
```

For time, specify a time in seconds. The default time is 600, which is 10 minutes. There is no minimum or maximum time.

rearch_no_release Directive: Inhibiting Releasing for Rearchived Files

By default, files marked for re-archiving are released. If the `rearch_no_release` directive is specified in the `releaser.cmd` file, the releaser does not release the files marked for re-archiving. This directive has the following format:

```
rearch_no_release
```

weight-age Directives: Specifying File Age-Related Priority Directives

Files are released from a file system according to the priority determined by directives defined in the releaser.cmd file. Both file age and file size are considered. By default, sites release the largest, oldest files first, leaving the smallest, newest files on disk. The following sections describe how the releaser considers a file's age and size when determining the release priority of files in a file system.

The releaser considers the following ages when determining a file's release priority:

- The age since it was last accessed
- The age since it was last modified
- The age since it changed residency in disk cache

By default, the age of a file is the more recent of the file's three ages. In some cases, an age derived from all three ages is preferred. In other cases, the access age of a file takes precedence over the modification age. You can use the following directives to control the age of a file when calculating the release priority for a file.

weight_age Directive

The weight_age directive specifies whether a file's default age is given a weighting factor. For float, specify a floating-point number from 0.0 through 1.0. By default, float = 1.0.

This directive cannot be used with the weight_age_access, weight_age_modification, or weight_age_residence directives.

weight_age_access, weight_age_modification, and weight_age_residence Directives

The weight_age_access, weight_age_modification, and weight_age_residence directives specify that a file's age be determined based on one, two, or three of the possible ages. For float, specify a floating-point number from 0.0 through 1.0. By default, float = 1.0.

These directives cannot be specified with the weight_age directive.

If the weight_age_access, weight_age_modification, and weight_age_residence directives are all used, the age-related priority for a file is calculated as follows and as shown in the following equation:

1. File age data is gathered for each file's possible age.
2. File age data is multiplied by the weighting factors specified in the releaser.cmd file.
3. The products of the multiplication are added together.

```
(file access age * weight_age_access) + (file modification age * weight_age_modification) + (file
residency age * weight_age_residence) = age_related_priority
```

Example - Ignoring Modification and Access Age

The following example specifies that only the file's residence age is considered when the release priority of a file is calculated.

```
weight_age_residence = 1.0
weight_age_modify = 0.0
weight_age_access = 0.0
```

weight-size Directive: Specifying File Size-Related Priority Directives

After a file's age-related priority is calculated, it is multiplied by the file's size-related priority. To determine the size-related component of a file's release priority, the releaser multiplies the size of the file (in 4-kilobyte blocks) by the weight specified for the weight_size directive. The format of the weight_size directive is as follows:

```
weight_size = <float>
```

For float, specify a floating-point number from 0.0 through 1.0. By default, float = 1.0.

Example - Ignoring the File Size

The following example specifies that the file size is to be ignored for all files in the samfs1 and samfs2 file systems.

```
# releaser.cmd file
logfile = /var/adm/default.releaser.log
weight_size = 0.0
#
fs = samfs1
weight_age = 1.0
logfile = /var/adm/samfs1.releaser.log
#
fs = samfs2
weight_age_modify = 0.3
weight_age_access = 0.03
weight_age_residence = 1.0
logfile = /var/adm/samfs2.releaser.log
```

Configuring the Stager

Contents

Configuring the Stager

This page shows how to configure the stager through the `stager.cmd` file.

About the stager.cmd File

In the the `stager.cmd` file, specify directives to override the default behaviors. You can configure the stager to stage files immediately, to never stage files, to staging partially, and to specify other staging actions. For example, specifying the never-stage attribute benefits applications that access small records from large files because the data is accessed directly from the archive media without staging the file online.

This section describes the stager directives. For additional information about stager directives, see the `stager.cmd`(4) man page. If you are using the SAM-QFS Manager software, you can control staging from the File System Summary or File System Details page. You can browse the file system and see the status of individual files, use filters to view certain files, and select specific files to stage. You can select which copy to stage from or let the system choose the copy.

Example **`stager.cmd`** File

The following example shows a `stager.cmd` file after all possible directives have been set.

```
# This is stager.cmd file /etc/opt/SUNWsamfs/stager.cmd
drives=dog 1
bufsize=od 8 lock
logfile=/var/adm/stage.log
maxactive=500
```

How to Create a stager.cmd File

1. In the `/etc/opt/SUNWsamfs/stager.cmd` file, add the directives to control staging at your site, according to the information in the following sections:
   - drives Directive: Specifying the Number of Drives for Staging
   - bufsize Directive: Setting the Stage Buffer Size
   - logfile Directive: Specifying a Log File
   - maxactive Directive: Specifying the Number of Stage Requests
2. Save and close the `stager.cmd` file.
3. Propagate the file changes and restart the system.

```
# samd config
```

drives Directive: Specifying the Number of Drives for Staging

By default, the stager uses all available drives when staging files. If the stager keeps all the drives busy, it can interfere with the archiver's activities. The `drives` directive specifies the number of drives available to the stager. This directive has the following format:

```
drives = <library> <count>
```

Argument	Meaning
library	The family set name of a library as it appears in the `mcf` file.
count	The maximum number of drives to be used. By default, this is the number of drives configured in the `mcf` file for this library.

You can also specify this directive by using the SAM-QFS Manager software. For more information, see the SAM-QFS Manager online help.

Example - Drives Directive

The following example specifies that only one drive from the dog family set's library is used for staging files:

```
drives = dog 1
```

bufsize Directive: Setting the Stage Buffer Size

By default, a file being staged is read into memory in a buffer before being restored from the archive media to disk cache. Use the `bufsize` directive to specify a buffer size and, optionally, to lock the buffer. These actions can improve performance. You can experiment with various buffer-size values. The directive has the following format:

```
bufsize = <media> <buffer-size> [lock]
```

Argument	Meaning
media	Specify the archive media type from the list on the `mcf`(4) man page.
buffer-size	A number from 2 through 8192. The default is 16. This value is multiplied by the dev*`blksize`* value for the media type, and the resulting buffer size is used. The dev*`blksize`* value is specified in the `defaults.conf` file. The higher the number specified for buffer-size, the more memory is used. For more information, see the `defaults.conf`(4) man page.

`lock`	The `lock` argument indicates that the stager should use locked buffers when staging archive copies. If `lock` is specified, the stager sets file locks on the stage buffer in memory for the duration of the copy operation. This avoids the overhead associated with locking and unlocking the buffer for each I/O request and can thereby result in a reduction in system CPU time.The `lock` argument should be specified only on large systems with large amounts of memory. Insufficient memory can cause an out-of-memory condition.The `lock` argument is effective only if direct I/O is enabled for the file being staged. By default, `lock` is not specified, and the file system sets the locks on all direct I/O buffers, including those for staging. For more information about enabling direct I/O, see the `setfa`(1) man page, the `sam_setfa`(3) library routine man page, or the `-O forcedirectio` option on the `mount_samfs`(1M) man page.

You can also specify this directive by using the SAM-QFS Manager software. For more information, see the SAM-QFS Manager online help.

logfile Directive: Specifying a Log File

You can request that the SAM-QFS software collect file-staging event information and write it to a log file. By default, no log file is written. The `logfile` directive specifies a log file to which the stager can write logging information. The stager writes one or more lines to the log file for each file staged. This line includes information such as the name of the file, the date and time of the stage, and the volume serial number (VSN). The directive has the following format:

```
logfile=<filename> [<event>]
```

Argument	Meaning
filename	Specify a full path name.
event	Specify one or more staging events. If you specify more than one event, use spaces to separate each them. Possible event specifications are as follows. `all` - Logs all staging events. `start` - Logs when staging begins for a file. `finish` - Logs when staging ends for a file. Enabled by default. `cancel` - Logs when the operator cancels a stage. Enabled by default. `error` - Logs staging errors. Enabled by default.

You can also specify this directive by using the SAM-QFS Manager software. For more information, see the SAM-QFS Manager online help.

Example: Specifying a Stager Log File

The following directive creates the `/var/adm/stage.log` file:

```
logfile=/var/adm/stage.log
```

Example: Stager Log File

```
S 2003/12/16 14:06:27 dk disk01 e.76d 2557.1759 1743132 /sam1/testdir0/filebu 1 root other root 0
-
F 2003/12/16 14:06:27 dk disk01 e.76d 2557.1759 1743132 /sam1/testdir0/filebu 1 root other root 0
-
S 2003/12/16 14:06:27 dk disk02 4.a68 1218.1387 519464 /sam1/testdir1/fileaq 1 root other root 0 -
S 2003/12/16 14:06:43 dk disk01 13.ba5 3179.41 750880 /sam1/testdir0/filecl 1 root other root 0 -
F 2003/12/16 14:06:43 dk disk01 13.ba5 3179.41 750880 /sam1/testdir0/filecl 1 root other root 0 -
S 2003/12/16 14:06:59 dk disk01 17.167b 1155.1677 1354160 /sam1/testdir0/filedb 1 root other root
0 -
F 2003/12/16 14:06:59 dk disk01 17.167b 1155.1677 1354160 /sam1/testdir0/filedb 1 root other root
0 -
S 2003/12/16 14:06:59 dk disk02 f.f82 3501.115 1458848 /sam1/testdir1/filecb 1 root other root 0 -
S 2003/12/16 14:07:15 dk disk01 1f.473 1368.1419 636473 /sam1/testdir0/fileed 1 root other root 0
-
S 2003/12/16 14:07:15 dk disk02 16.f15 3362.45 1065457 /sam1/testdir1/filecz 1 root other root 0 -
S 2003/12/16 14:07:31 dk disk01 23.201d 3005.1381 556807 /sam1/testdir0/fileeq 1 root other root 0
-
S 2003/12/16 14:07:47 dk disk01 26.c4d 2831.1113 1428718 /sam1/testdir0/fileez 1 root other root 0
-
S 2003/12/16 14:07:47 dk disk02 1b.835 3736.59 1787855 /sam1/testdir1/filedp 1 root other root 0 -
```

The following describes the content of the stager log file fields.

Field	Example Value	Content Description
1	S	Stage activity - S for start, C for canceled, E for error, F for finished.
2	2003/12/16	Date of the stage action, in yyyy/mm/dd format.
3	14:06:27	Time of the stage action, in hh:mm:ss format.
4	dk	Archive media type. For information about media types, see the `mcf`(4) man page.
5	disk01	VSN.
6	e.76d	Using hexadecimal format, the physical position of the start of the archive file on media (`tar`(1) file) and the file offset on the archive file.
7	2557.1759	Inode number and generation number. The generation number is used in addition to the inode number for uniqueness because inode numbers are reused.
8	1743132	Length of the file.
9	`/sam1/testdir0/filebu`	Name of the file.
10	1	Archive copy number.
11	root	User ID of the file.
12	other	Group ID of the file.
13	root	Group ID of the requestor.
14	0	Equipment ordinal of the drive from which the file was staged.
15	-	A V in this field indicates that data verification is being used for the file.

maxactive Directive: Specifying the Number of Stage Requests

The `maxactive` directive enables you to specify the number of stage requests that can be active at any one time. The directive has the following format:

```
maxactive=<number>
```

By default, number is 4000. The minimum number allowed is 1. The maximum allowed is 500,000.

Example - Number of Stage Requests

The following example specifies that no more than 500 stage requests can be in the queue simultaneously:

```
maxactive=500
```

Archive Set Assignment Directive: Specifying Stage Attributes for All Files in an Archive Set

Most directives in the archiver.cmd file affect only archiving, but you can use the archive set assignment directive to specify stage attributes that apply to all files in an archive set.

The Archive Set Directives (archiver.cmd) page describes the archive set assignment directive and its arguments completely. The following table shows the staging directives that can appear in an archive set assignment directive in the archiver.cmd file.

Directive	Effect
-stage a	Specifies that the files in the archive set should be associatively staged.
-stage d	Reset to default.
-stage n	Specifies that the files in the archive set should never be staged.

Prioritizing Preview Requests

Both the archiver and stager processes request that media is loaded and unloaded. If the number of requests exceeds the number of drives available for media loads, the excess requests are sent to the preview queue.

The number of entries that can be in the preview queue is determined by the previews= directive in the defaults.conf file. For information about changing the value of this directive, see the defaults.conf(4) man page.

By default, preview requests are satisfied in first-in-first-out (FIFO) order.
The overall priority of preview requests is determined by the combination of static and dynamic factors. Higher numbers correspond to higher priority. A static priority factor is set when the request is generated. Its effect does not change the overall priority after the request is generated and is waiting to be satisfied. A dynamic priority factor can increase or decrease the overall priority of a request while the request is waiting to be satisfied.

You can override the FIFO default by entering directives in the /etc/opt/SUNWsamfs/preview.cmd command file.

About the preview.cmd File

The sam-amld daemon reads the preview.cmd file at startup. This file orders the requests in the preview queue according to whether the request is for staging or archiving. You can increase the priority for specific VSNs and you can control the priority of preview requests for specific file systems.

The following rules apply to the preview.cmd file:

- Place one directive per line.
- If you change this file while the sam-amld daemon is running, restart the daemon to have your changes take effect.
- Begin comment lines begin with a pound sign (#).
  For more information about this file, see the preview.cmd(4) man page.

The following types of directives are used in the preview.cmd file:

- Global directives, which apply to all file systems
- File-system directives, specific to individual file systems
  Global directives are placed at the top of the file and their settings apply to all file systems.
  File system directives begin with the fs = directive, which names the file system to which all subsequent directives apply. More than one block of file directives can appear in a file. File system directives apply until the next fs = line is encountered or until the end of file is encountered.

When multiple directives affect a file system, the directives that are specific to that file system override the global directives.

In the following sections, edit the preview.cmd file to control the preview queue, according to the information in the following section:

- How to Set the Global VSN and Age Priority Directives
- How to Set Global or File-System-Specific Water Mark Directives

* Setting Up a Preview Request Priority Scheme

How to Set the Global VSN and Age Priority Directives

> Info -
> The VSN and age priority directives are global directives, so they are placed before any file-system-specific directives in the `preview.cmd` file.

1. Update the `vsn_priority` directive.

```
vsn_priority = <value>
```

   This directive is a static priority factor that indicates the value by which the total priority increases when there is a high-priority volume. The default value for is `1000.0`. To use this priority factor, a volume must have its priority flag set before it is scheduled as a preview request. Use the `chmed`(1M) command to set the priority flag with the p option (for example, `chmed +p lt.`AAA123).

2. Update the `age_priority` directive.

```
age_priority = <factor>
```

   This directive is a static priority factor, although its effect is dynamic. This factor is multiplied by the number of seconds for which a request is a preview request. The result is then added to the overall priority of the request. The longer a request waits, the higher the priority becomes. Setting this factor ensures that older requests are not indefinitely superseded by newer requests with other higher-priority factors.
   Setting this factor to more than 1.0 increases the importance of the time factor in calculation of the total priority and setting it to less than 1.0 decreases the importance of the time factor. Setting the factor to 0.0 eliminates the time factor from the overall priority calculation.
   A volume whose priority flag is not set increases in priority based on the time it remains in the queue. Its priority can become higher than a VSN that comes into the queue later with the priority flag already set.

How to Set Global or File-System-Specific Water Mark Directives

The water mark preview request directives can be used as either global or file-system-specific directives. These directives determine the water mark priority of the preview requests, as shown in the following equation.

`lwm_priority` + `lhwm_priority` + `hlwm_priority` + `hwm_priority` = water mark priority

Together, the four water mark settings create a dynamic priority factor that includes a percentage value indicating how full the file system is and the levels at which the HWM and LWM are set. The value assigned to a preview request is determined by whether a factor is global, specific to a file system, or not set.

The water mark priorities are used to calculate only requests for archiving. They are not used to calculate media requests for staging. When the water mark priority factor is a positive number, the result on the overall calculated priorities is to raise archiving requests over staging requests. In contrast, when the water mark priority factor is a negative number, the overall priority for archiving requests is reduced, which tends to favor staging requests over archival requests. A water mark priority factor of 0.0 (or no specified directive) indicates that no special action occurs.

For more information, see Example 1 - Scheme for Enforcing Stage Requests.

The water mark directives have the following format:

```
<wmtype>_priority = <value>
```

Water Mark Directive	Argument
`lwm_priority` = value	Specify the amount by which you want the water mark priority factor to change for archiving requests when the file system is below the LWM level. The default is 0.0.

`lhwm_priority` = value	Specify the amount by which you want the water mark priority factor to change for archiving requests when the file system crosses from below to above the LWM level but remains below the HWM level. This shift indicates that the file system is filling up. The default is 0.0.
`hlwm_priority` = value	Specify the amount by which you want the water mark priority factor to change for archiving requests when the file system has crossed from above the HWM level to below it, but remains above the LWM level. This shift indicates that the releaser was not able to free enough disk space to leave the file system below the LWM level. The default is 0.0.
`hwm_priority` = value	Specify the amount by which you want the water mark priority factor to change for archiving requests when the file system is above the HWM level. The default is 0.0.

When a file system crosses from one condition to another, the priority of each volume associated with that file system is recalculated based on the appropriate water mark priority setting, with or without the `chmed`(1M) command's p option.

Example: Settings for a File System Below the LWM Level

The following example frees enough disk space so that the file system goes below the LWM level.

```
lhwm_priority = -200.0
hlwm_priority = 100.0
```

Setting Up a Preview Request Priority Scheme

The total priority for a preview request is the sum of all priority factors:

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

Change the default FIFO scheme only for reasons, such as the following:

- Ensure that staging requests are processed before archive requests.
- Ensure that archive requests gain top priority when a file system is about to fill up.
- Push requests that use a specific group of media to the top of the preview request list.
  The following example shows a `preview.cmd` file that addresses these three conditions.

Example - Sample **`preview.cmd`** File

```
# 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
```

For environments in which user access to data is of paramount importance, the VSN drives are limited, or file archiving is performed as a background function, use the `preview.cmd` file to influence how the storage system resources handle staging requests. You can customize the settings in the `preview.cmd` file to support any of the preceding scenarios and influence the configured SAM-QFS environment.

Because data is not affected by the settings in this file, you are encouraged to experiment and adjust the directive settings to achieve the proper balance between archiving and staging requests when weighed against the priorities of each preview request.

Example 1 - Scheme for Enforcing Stage Requests

The following example calculations show how you can use a negative value for `wm_priority` to ensure that stage requests have priority over archive requests. This example assumes the following:

- Several requests are sitting in the queue for 100 seconds.
- The default value `vsn_priority` is 1000.

The following table shows how the total request priorities are calculated.

Priority	Calculation
Archive VSN with priority, LWM:	1000 + (-200) + (1 x 100) = 900
Stage VSN with priority, LWM:	1000 + 0 + (1 x 100) = 1100
Stage VSN without priority, LWM:	0 + 0 + (1 x 100) = 100

Example 2 - Scheme for Enforcing Archive Requests

When the environment is balanced between the importance of staging a file for the user and the importance of getting new files archived to media, the biggest concern is exceeding the HWM level. In this situation, if not enough files have met their archive requirements to lower the percentage of the file system that is full, meeting the pending archive requests is the best way to keep the file system from reaching its limit. In this situation, the `preview.cmd` file can be as simple as the following:

```
hwm_priority = 500.0
```

Example 3 - Scheme for Ranking Requests by Media Type

A site has an environment in which users are working on groups of files that use specific volumes and are segregated from other users. In this environment, certain projects might have higher priorities at certain times; therefore, greater priority is required from the available system storage resources. The following example gives users and their media the appropriate priority:

```
hwm_priority = 5000.0
```

Then, for every volume in the priority user's group, include the following information:

```
# chmed +p lt. <VSN>
```

Now every request that requires the specified VSN is placed above other pending mount requests in the preview queue.
Later, to lower the priority of the user's media, include the following reverse command for every VSN:

```
# chmed -p lt. <media-type>
```

> Note -
> A request for a select group of VSNs always takes precedence in the preview request queue if the `chmed(1M)` command's p flag is set.

Example 4 - Scheme for Complex Priorities

Assume that there are two file systems with the following requirements:

- No request must wait too long in the queue (`age_priority`).
- When one of the file systems is below the LWM level, staging requests take precedence.
- When one of the file systems is above the LWM level but below the HWM level, do not prioritize requests.
  The following example shows the affected directives.

  ```
  lwm_priority = -200.0
  lhwm_priority = 0.0
  hlwm_priority = 0.0
  ```

  When one of the file systems goes over the HWM level, archive requests take priority.

Assume both file systems are over the HWM level but it is important to prevent the second file system (`samfs2`) from reaching its limit. The following example shows a `preview.cmd` file that prioritizes requests according to the requirements in the preceding list.

```
age_priority = 100.0
vsn_priority = 20000.0
lhwm_priority = -200.0
hlwm_priority = -200.0
fs = samfs1
hwm_priority = 1000.0
fs = samfs2
hwm_priority = 5000.0
```

Creating Parameters Files for Network Attached Automated Libraries

Contents

Creating Parameters Files for Network Attached Automated Libraries

Perform the applicable procedure in this section if you have network attached automated libraries in your Sun Storage Archive Manager (SAM) environment. You must configure your storage devices first, as described in Configuring Storage Devices for Archiving.

You can include automated libraries in a SAM environment either by directly attaching them to the server or by attaching them to the environment's network. Libraries attached through a SCSI or Fibre Channel (FC) attachment are direct attached libraries. Libraries attached through a network attachment are network attached libraries. In this task, you create a parameters file for each network attached library to be included in your environment.

> Note -
> The examples and the discussions in the following sections mention both the parameters files for network attached automated libraries and the `mcf` file. The `mcf` file is the main configuration file for the SAM software. You created your `mcf` file in Configuring the File System Environment. The parameters file and the `mcf` file are both mentioned in this section because the two files reference each other.

How To Configure an ADIC/Grau Automated Library Parameters File

The ADIC/Grau automated library operates within SAM environments through the `grauaci` interface. This interface uses the DAS/ACI 3.12 interface supplied by ADIC/Grau. For more information about the DAS/ACI interface, see your ADIC/Grau documentation.

> Note -
> ADIC/Grau network attached libraries are not supported by the SAM software on an x64 hardware platform.

Before You Begin

Ensure that the following are true and the ADIC/Grau automated library is prepared for inclusion in a SAM environment.

- The ADIC/Grau automated library is operational.
- The ADIC/Grau library is operating on the DAS (Distributed AML Server).
- Both the `avc` (avoid volume contention) and the `dismount` parameters are set to `true` in the DAS configuration file for this client.

Follow this procedure to create a parameters file for each ADIC/Grau library that you want to configure.

1. Change to the `/etc/opt/SUNWsamfs` directory.
   The parameters file can be written to any directory, but the most common location is `/etc/opt/SUNWsamfs`.

Note -
When you created your mcf file, in Configuring the File System Environment, you included the full path name to the parameters file in the mcf file. Make sure that the mcf file points to the correct location for the parameters file that you create in this procedure.

2. Open a new file abd give it a name that corresponds to the library that you are configuring. For example:

```
# vi grau50
```

3. Type a list of keyword = value parameter lines in the ADIC/Grau parameters file.
   The various keyword values identify the ADIC/Grau automated libraries, the drives associated with the libraries, and the server name. All keyword and value entries are case-sensitive, so enter them exactly as specified in the DAS configuration file and in the SAM mcf file.
   The following table shows the keyword = value parameters that must appear in the ADIC/Grau parameters file.

Parameter	Meaning
client = client-id	The name of the client as defined in the DAS configuration file. This parameter is required.
server = server-id	The host name of the server running the DAS server code. This parameter is required.
acidrive drive-id = path	The name of the drive as configured in the DAS configuration file. For path, specify the path to the drive as entered in the Equipment Identifier field of the SAM mcf file. Include one acidrive line for every drive assigned to the client.

   Comments can appear anywhere on any line, and they must begin with a pound sign (#). The system ignores characters to the right of the pound sign.

If the ADIC/Grau library contains various media types, a media changer exists for each media type. Each media changer has a unique client name in the DAS configuration, a unique library catalog, and a unique parameters file.

The sample ADIC/Grau parameters file shown in the following example defines one ADIC/Grau automated library supporting DLT tape and one ADIC/Grau automated library supporting a Hewlett-Packard optical drive.

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

Note -
Configuration Example for a Shared File System on a Solaris OS Platform shows the mcf file that corresponds with the example of a ADIC/Grau network attached automated library parameters file that was created in this procedure. The example mcf file points to the grau50 file in the /etc/opt/SUNWsamfs directory.

The following directory contains diagnostic information that can be useful for troubleshooting:

/var/opt/SUNWsamfs/.grau

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

How To Configure an IBM 3494 Automated Library Parameters File

The IBM 3494 automated tape library operates in SAM environments with the assistance of the IBM `lmcpd` daemon package. You can obtain the IBM `lmcpd` daemon package from IBM.

> Note -
> IBM 3494 network attached libraries are not supported by SAM software on an x64 hardware platform.

Before You Begin

Ensure that the following are true and the IBM 3494 automated library is prepared for inclusion in a SAM environment.

- The IBM 3494 automated library is operational.
- The IBM `lmcpd` daemon package is installed and working.
- The `/etc/ibmatl.conf` file is configured and working.
- The IBM 3494 automated library can be used as a single physical library or as multiple logical libraries. If you divide this library into multiple logical libraries, create a parameters file for each logical library.

Follow this procedure to create a parameters file for each physical or logical library that you want to include in the SAM environment.

1. Change to the `/etc/opt/SUNWsamfs` directory. The parameters file can be written to any directory, but the most common location is `/etc/opt/SUNWsamfs`.

   > Note -
   > When you created your `mcf` file in Configuring the File System Environment, you included the full path name to the parameters file in the `mcf` file. Make sure that the `mcf` file points to the correct location for the parameters file that you create in this procedure.

2. Create a new file and give it a name that corresponds to the library that you are configuring. For example:

```
# vi ibm50
```

3. Type a list of keyword = value and pathname = value pairs in the IBM 3494 parameters file.

All arguments are case-sensitive. The following table shows how to specify the parameters.

Parameter	Meaning
`name` = name	The name assigned by you as system administrator, and specified in the `/etc/ibmatl.conf` file. This name is also the symbolic name of the library. This parameter must be supplied. There is no default.
`category` = hexnumber	A hexadecimal number between `0x0001` and `0xfeff`. By default, the SAM software sets this value to 4 for media under its control. If you have divided your physical library into multiple logical libraries, make sure that the value of hexnumber is different in each logical library. This parameter determines which tapes are assigned to which library. When you import media into the library, they are added to the catalog, and their `category` = value is changed to the value specified by this `category` = hexnumber parameter.
`access` = permission	Either `shared` or `private`. • Specify `private` if you are using the library as one physical library. This is the default. • Specify `shared` if you are dividing the library into multiple logical libraries.
device-pathname = device-number	For device-pathname, specify the path of a drive. You must have a device-pathname entry for every drive in the library attached to this machine, and each device-pathname entry must match the Equipment Identifier value of the corresponding entry in the `mcf` file. For device-number, the device number as described in the IBM documentation. You can obtain this number by running the IBM `mtlib` utility.

Comments can appear anywhere on any line and must begin with a pound sign (#). The system ignores characters to the right of the pound sign.

The following example shows a sample `/etc/ibmatl.conf` file. Information for this file was obtained from the `mtlib` utility supplied by IBM.

```
#
# This is file: /etc/ibmatl.conf
# Set this file up according the documentation supplied by IBM.
3493a   198.174.196.50  test1
```

After the lmcpd daemon is running, you can use the IBM mtlib utility to obtain the device numbers. The following example shows output from mtlib.

```
# mtlib -l 3493a -D
0, 00145340 003590B1A00
1, 00145350 003590B1A01
```

The following example shows a sample parameters file for an IBM 3494 library.

```
#
# This is file: /etc/opt/SUNWsamfs/ibm50
#
name = 3493a    # From /etc/ibmatl.conf
/dev/rmt/1bn = 00145340     # From mtlib output
/dev/rmt/2bn = 00145350     # From mtlib output
access=private
category = 5
```

Note -
Configuration Example for a Shared File System on a Solaris OS Platform shows the mcf file that corresponds with the IBM 3494 network attached automated library parameters file that was created in this procedure. The example mcf file points to file ibm50 in the /etc/opt/SUNWsamfs directory.

How To Configure a Sony Network Attached Automated Library Parameters File

The Sony network attached automated library operates within the SAM environment through the DZC-8000S Application Interface Library package. This software package provides the application programming interface (API) to the PetaSite Controller (PSC). For more information about the DZC-8000S interface, see the Sony PetaSite Application Interface Library DZC-8000S, which is available from Sony.

Note -
Sony network attached libraries are not supported by SAM software on an x64 hardware platform.

Note -
The information in this section applies only to Sony automated libraries that are network attached through a Sony DZC-8000S interface. Sony direct attached B9 and B35 automated libraries or Sony direct attached 8400 PetaSite automated libraries do not require a parameters file.

Before You Begin

Ensure that the following are true and the Sony network attached automated library is prepared for inclusion in a SAM environment.

- The Sony network attached automated library is operational.
- The Sony PSC configuration file is installed and working.

Follow this procedure to create a parameters file for each Sony network attached library that you want to configure.

1. Change to the /etc/opt/SUNWsamfs directory.
   The parameters file can be written to any directory, but the most common location is /etc/opt/SUNWsamfs.

> **Note -**
> When you created your mcf file in Configuring the File System Environment, you included the full path name to the parameters file in the mcf file. Make sure that the mcf file points to the correct location for the parameters file that you create in this procedure.

2. Create a new file and give it a name that corresponds to the library that you are configuring. For example:

```
# vi sonyfile
```

3. Type a list of keyword = value parameter lines in the Sony parameters file.

The various keyword values identify the Sony automated libraries, the drives associated with the libraries, and the host name. All keyword and value entries are case-sensitive, so type them exactly as specified in the configuration file and in the SAM mcf file.
The following table shows the keyword = value parameters that must appear in the Sony parameters file. All parameters are required.

Parameter	Meaning
userid = user-id	A number from 0 to 65535, inclusive. If you specify a number other than 0, it must be the PSC ID. The userid parameter identifies the user during initialization of the PetaSite automated library functions.
server = server-id	The host name of the server running the PSC server code.
sonydrive drive-id = path	For drive-id, the drive bin number as configured in the PSC configuration file. Include one sonydrive line for every drive defined in the mcf file. For path, the path to the drive as entered in the Equipment Identifier field of the SAM mcf file.

Comments can appear anywhere on any line, but they must begin with a pound sign (#). The system ignores characters to the right of the pound sign.

The following example shows a parameters file for a Sony network attached automated library.

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

> **Note -**
> Configuration Example for a Shared File System on a Solaris OS Platform shows the mcf file that corresponds with the Sony network attached automated library parameters file that was created in this procedure. The example mcf file points to file sonyfile in the /etc/opt/SUNWsamfs directory.

How To Configure a StorageTek ACSLS-Attached Automated Library Parameters File

In many respects, the way in which SAM systems interoperate with StorageTek ACSLS-attached automated libraries is very similar to the way in which they interoperate with direct attached automated libraries. However, the installation and configuration procedure of a StorageTek

ACSLS-attached automated library requires additional steps.

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

> Note -
> The SAM-QFS Manager software supports the automatic discovery and configuration of ACSLS network attached libraries. You do not need to set up the parameters file before configuring the library in SAM-QFS Manager. For more information, see the SAM-QFS Manager online help.

Before You Begin

Ensure that the following are true and the StorageTek ACSLS-attached automated library is prepared for inclusion in a SAM environment.

- The StorageTek ACSLS automated library is operational.
- The StorageTek ACSLS software package is installed and working.

Follow this procedure to create a parameters file for each StorageTek ACSLS-attached library that you want to configure. See the `stk`(7) man page for more information.

1. Change to the `/etc/opt/SUNWsamfs` directory.
   The parameters file can be written to any directory, but the most common location is `/etc/opt/SUNWsamfs`.

   > Note -
   > When you created your `mcf` file in Configuring the File System Environment, you included the full path name to the parameters file in the `mcf` file. Make sure that the `mcf` file points to the correct location for the parameters file that you create in this procedure.

2. Create a new file and give it a name that correspond to the library that you are configuring. For example:

```
# vi stk50
```

3. Type a list of keyword = value parameter lines in the StorageTek parameters file.

The following table shows the keywords to use.

Parameter	Meaning
`access` = userid	(Optional) The user identification value used by the StorageTek software for access control. If the `access` parameter is not supplied, the access control string is a null string, indicating that there is no userid.
`hostname` = hostname	The host name of the server running the StorageTek ACSLS interface.
`portnum` = portnum	The port number used for communication between ACSLS and the SAM software. For information about the portnum argument, see the `stk`(7) man page.
`ssihost` = hostname	The name of the SAM server on the LAN that connects to the ACSLS host. Specify this directive only if you are including a multihomed SAM server in your environment. The default is the name of the local host.
`ssi_inet_port` = ssi-inet-port	The fixed port number for incoming responses and specifies the port the SSI will use for incoming ACSLS responses in a firewall environment. Specify either 0 or a value from 1024 to 65535. Setting this environmental variable to a non-zero value forces SSI to use this port for incoming ACSLS responses.
`csi_hostport` = csi-port	The port on the ACSLS server to which the StorageTek SSI daemon is to send its ACSLS requests. Specify either 0 or a value from 1024 to 65535, inclusive. Setting this variable to 0 or leaving it unset causes the system to query the port mapper on the ACSLS server.

capid = (acs = acsnum, lsm = lsmnum, cap = capnum)	The cartridge access point (CAP), in terms of the StorageTek library, to be used when the `export(1M) -f` command is specified. The `capid` description starts with an open parenthesis followed by three keyword = value pairs followed by a closing parenthesis. Use a comma (as shown), a colon, or a space to separate the keyword = value pairs. For acsnum, specify the asynchronous communications server (ACS) number for this CAP as configured in the StorageTek library. For lsmnum, specify the length subnet mask (LSM) number for this CAP as configured in the StorageTek library. For capnum, specify the CAP number for this CAP as configured in the StorageTek library.
capacity = (index = value, [index = value]...)	The capacities of the supported cartridges. Use a comma to separate the index = value pairs, and enclose the string in parentheses. For index, specify the index of the supplied `media_type` file, located in the following ACSLS directory: `/export/home/ACSSS/data/internal/mixed_media/media_types.dat`. For value, specify the capacity of the cartridge type in units of 1024 bytes. In general, supplying a capacity entry is necessary only for an index of new cartridge types or to override the supported capacity.
device-path-name = (acs = value, lsm = value, panel = value, drive = value) [shared]	The path to the device on the client. Specify one device-path-name = entry for each drive attached to this client. This parameter describes the drive within the StorageTek automated library. This description starts with an open parenthesis followed by four keyword = value pairs and a closing parenthesis. Use a comma (as shown), a colon, or a space to separate the keyword = value pairs. The `shared` keyword is optional. It specifies that the drive can be shared between two or more SAM processes from two or more hosts. For more information about implementing shared drives, see About Shared Drives or see the `stk(7)` man page. For the value arguments, use the information supplied by the ACSLS query drive command. The value specifications are: • `acs` - ACS number for the drive as configured in the StorageTek library • `lsm` - LSM number for the drive as configured in the StorageTek library • `panel` - PANEL number for the drive as configured in the StorageTek library • `drive` - DRIVE number for the drive as configured in the StorageTek library

The following example shows a parameters file for a StorageTek ACSLS-attached automated library.

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

Note -
Configuration Example for a Shared File System on a Solaris OS Platform shows the `mcf` file that corresponds with the StorageTek ACSLS-attached automated library parameters file that was created in this procedure. The example `mcf` file points to file `stk50` in the `/etc/opt/SUNWsamfs` directory.

About Shared Drives

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

The shared-drives capability enables two or more `mcf` files to define the same drive, making the drive available to multiple SAM processes. However, these multiple processes cannot share media. Each SAM process must still maintain its own set of VSNs.

The shared-drives feature can be useful, for example, if a library is attached to more than one host system in a SAM environment. The SAM processes coordinate the use of a drive and keep the drives in a library busy.

You can configure some network attached libraries to share one or all media drives between multiple SAM processes on multiple host systems. All of the StorageTek ACSLS-attached libraries support shared drives in SAM environments.

How To Create a Shared Drive for Multiple SAM Processes

- To implement one or more shared drives, specify the `shared` keyword in the parameters file for each drive that is to be shared. The placement of the `shared` keyword is specific to each manufacturer's library, so see the vendor-specific sections for more information.

How To Change the Idle Time for a Shared Drive

By default, a cartridge in a shared drive can be idle for 60 seconds before being unloaded. To change this timing, change the `shared_unload` value in the `defaults.conf` file to the new value (in seconds).

Managing Automated Libraries and Manually Loaded Drives

Contents

Managing Automated Libraries and Manually Loaded Drives

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

This page describes aspects of using automated libraries and manually loaded drives in a Sun Storage Archive Manager (SAM-QFS) environment. In addition, this page describes the operator-oriented load notification facility that alerts an operator when a requested volume is not in a library.

> Note -
> The SAM-QFS software interoperates with automated libraries from many manufacturers. Contact Sun Customer Support for information pertinent to library model numbers, firmware levels, and other compatibility information.

Vendor-Specific Library Operational Procedures

Certain operations for some automated libraries may differ from those described in this chapter. To determine whether your automated library has additional vendor-specific operating instructions in a SAM-QFS environment, see the Managing Vendor-Specific Libraries page.

How To Start Removable Media Operations

Typically, removable media operations commence when a file system is mounted.

- To start removable media operations manually, without mounting any file systems, issue the following command:

  ```
  # samd start
  ```

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

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

How To Stop Removable Media Operations

You can stop removable media operations and leave the file system mounted. You might do this, for example, if you want to manipulate cartridges manually.
Use the following procedure to stop operations:
#Issue the `idle` command to enable the archiver, stager, and other processes to complete current tasks. You can also idle drives by using the `samu`(1M) operator utility or by using SAM-QFS Manager.

```
samcmd aridle
samcmd stidle
```

Monitor the tape drive activity with the samcmd r command:

```
samcmd r
```

Wait to all the tape drives have stopped. Then unload the tape drives:

```
samcmd unload <eq>
```

For eq, provide the equipment number of the drive being unloaded as defined in the `mcf` file.

1. To unload multiple drives, issue a `samcmd unload` for each drive.
2. When the drive is empty, issue the following command:

   ```
   samd stop
   ```

> Note -
> Failing to follow the above procedures could result in tape media issues.
> You must issue the `samd`(1M) `unload` command for each idle tape drive. If you attempt to restart an idle drive without unloading it, unpredictable events occur when archiving, staging, and other activities are resumed.

When you restart operations, pending stages are reissued and archiving is resumed.

How To Turn On an Automated Library

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

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

- Use the following command to start an automated library:

```
samcmd on <eq>
```

For eq, specify the equipment number of the automated library as defined in the `mcf` file.
You can also perform this task by using `samu`(1M) or SAM-QFS Manager.

How To Turn Off an Automated Library

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

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

- Use the following command to turn off an automated library:

```
samcmd off <eq>
```

For eq, specify the equipment number of the automated library being addressed as defined in the `mcf` file.
You can also perform this task by using `samu`(1M) or SAM-QFS Manager.

How To Load a Cartridge Manually

When a cartridge is loaded, it moves from a storage slot to a drive and it is made ready to receive data. A cartridge is loaded automatically when a volume serial name (VSN) is requested for archiving or staging. You can load a cartridge at any time by issuing one of the following commands. You might do this, for example, during a disaster recovery operation or to analyze a tape.

- ```
  samcmd load <eq>:<slot>[:<partition>]
  ```
- ```
  samcmd load <media-type>.<vsn>
  ```

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

You can also perform this task by using `samu`(1M) or SAM-QFS Manager.
When you load a cartridge manually, it is loaded in the next available drive in the library. If you want to make a drive unavailable for this purpose, use the `samu`(1M) utility's `:unavail` command or change the state of the device using SAM-QFS Manager.

> Note -
> SAM-QFS does not support mixed media in direct attached libraries. If the library is partitioned, each partition must contain only one media type.

How To Unload a Cartridge Manually

When a cartridge is unloaded, it is removed from a drive. Unloading occurs automatically when a volume is no longer needed. You can unload a drive at any time, even if the drive is in `unavail` status, using the following command.

```
samcmd unload <eq>
```

For eq, specify the equipment number of the drive as defined in the `mcf` file.
You can also perform this task by using `samu`(1M) or SAM-QFS Manager.

Labeling and Unlabeling Cartridges

If you have stand-alone tape or optical devices, or if your automated library has no barcode reader, you must label cartridges as described in this section. To label cartridges, use the `tplabel`(1M) command for tapes or use the `odlabel`(1M) command for optical disks. These commands create a cartridge label that the SAM-QFS software can read.

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

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

```
labels = barcodes_low
```

When the software loads a barcoded cartridge for a write operation, it writes a label on the cartridge before the write begins. The cartridge must be write enabled, be unlabeled, and have a readable barcode.

How To Label or Relabel a Tape

Caution -
Labeling and relabeling a cartridge makes the data currently on the cartridge inaccessible to any software. Relabel a cartridge only if you are certain that you do not need the data that is stored on the cartridge.

- To label a new tape, use the following `tplabel`(1M) command.

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

- To relabel an existing tape, use the following `tplabel`(1M) command.

```
tplabel -old <vsn> -vsn <vsn> <eq>:<slot>
```

Argument	Meaning
vsn	Volume serial name. If you are relabeling, the new VSN can be identical to the old VSN.
eq	The equipment number of the drive as defined in the `mcf` file.
slot	The number of the tape's storage slot as recognized in the library catalog. This argument is not applicable for manually loaded drives.

After the command to label or relabel a tape is issued, the tape is loaded and positioned, and the tape label is written.

You can also perform this task by using SAM-QFS Manager.

Example

```
# tplabel -vsn TAPE01 -new 50:0
```

How To Label or Relabel an Optical Disk

- To label a new optical disk, use the following `odlabel`(1M) command.

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

- To relabel an existing optical disk, use the following `odlabel`(1M) command.

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

Argument	Meaning
vsn	Volume serial name. If you are relabeling, the new VSN can be identical to the old VSN.
eq	The equipment number of the drive as defined in the `mcf` file.
slot	The number of the disk's storage slot as recognized in the library catalog. This argument is not applicable to manually loaded drives.
partition	A side of a magneto-optical disk. The partition value must be `1` or `2`. This argument is not applicable to tape cartridges.

After the command to label or relabel an optical disk is issued, the optical disk is loaded and positioned, and the optical disk label is written.

You can also perform this task by using SAM-QFS Manager.

Example

```
# odlabel -vsn OPTIC01 -new 30:1:1
```

How To Audit a Volume

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

- Use the following command to update the amount of remaining space:

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

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

For more information, see the auditslot(1M) man page.
You can also perform this task by using the samu(1M) utility's `:audit` command or by using SAM-QFS Manager.

How To Audit a Direct Attached Automated Library

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

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

- To perform a full audit on a direct attached automated library, use the following command:

```
samcmd audit <eq>
```

For eq, specify the equipment number of the automated library as defined in the `mcf` file.
You can also perform this task by using the samu(1M) utility's `:audit` command or by using SAM-QFS Manager.

Using a Cleaning Cartridge

The SAM-QFS environment supports the use of cleaning tapes if cleaning tapes are supported by the hardware. If a tape drive requests cleaning, the system automatically loads a cleaning tape.

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

Cleaning practices differ from manufacturer to manufacturer. See Managing Vendor-Specific Libraries to determine whether specialized procedures are recommended for your equipment.

How To Use a Cleaning Cartridge With a Barcode

If the cleaning cartridge is barcoded, you can import it using the `samimport`(1M) command. This command moves the cartridge from the mailbox to a storage slot and updates the library catalog. In addition, the cleaning media flag is set, and the access count is set to the appropriate number of cleaning cycles, based on the media type. Each time the cartridge is used to clean a drive, the access count is decremented.

1. Make sure that the cleaning cartridge has a barcode of `CLEAN` or starts with the letters `CLN`.
2. Use the following command to import the cleaning cartridge into the automated library.

```
samimport <eq>
```

For eq, specify the equipment number of the automated library as defined in the `mcf` file.
You can also perform this task by using samu(1M) or SAM-QFS Manager.

How To Use a Cleaning Cartridge Without a Barcode

If the cartridge is not barcoded, it is not identified as a cleaning cartridge. After you import the cartridge, you must identify it.

1. Import the cartridge into the automated library.

```
samimport <eq>
```

For eq, specify the equipment number of the automated library as defined in the `mcf` file.

2. Change the type to a cleaning cartridge.

```
chmed +C <eq>:<slot>
```

For eq, specify the equipment number of the automated library.
For slot, specify the slot in which the cleaning cartridge has been loaded.
In the following example, the automated library is equipment number 50 and the cleaning cartridge is in slot 77:

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

3. Set the cleaning cycle count.

```
chmed -count <count-number> <eq>:<slot>
```

For eq, specify the equipment number of the automated library.
For slot, specify the slot in which the cleaning cartridge has been loaded.
The following example command sets the cleaning count on the cartridge to 20.

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

How To Reset the Number of Cleaning Cycles

Cleaning cartridges are useful for a limited number of cleaning cycles. The SAM-QFS system ejects the cartridge when the number of remaining cycles equals zero. Each time a cleaning tape is imported, the cleaning cycle is reset to the highest number of cycles for that type of tape. For example, a DLT cleaning tape has 20 cycles and an Exabyte cleaning tape has 10 cycles. You can view the number of remaining cycles with the `samu`(1M) utility's `:v` display or by using SAM-QFS Manager.

If automatic cleaning is available but all cleaning cartridges in the automated library have a cleaning cycle count of zero, the drive state is set to `off` and a message is issued in the SAM-QFS log.

- To reset the cycles on a cleaning tape:

```
chmed -count <count> <media-type>.<vsn>
```

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

How To Limit the Number of Cleaning Cycles

Certain drive errors can result in the repeated loading of cleaning cartridges until all cleaning cycles are consumed.

- To limit the number of cleaning cycles on cleaning cartridges:

```
chmed -count <count-number> <eq>:<slot>
```

  For eq, specify the equipment number of the automated library.
  For slot, specify the slot in which the cleaning cartridge has been loaded.

For example, the following command sets the cleaning cycle count to 20 for the cleaning cartridge in slot 77 of the automated library with the equipment number of 50.

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

How To Clean a Tape Drive Automatically

Beginning with the Sun Storage Archive Manager 4.4 version, the default setting for software-initiated tape drive cleaning is `off`. You can enable automatic cleaning in one of the following ways:

- Use the media changer's auto-cleaning feature, which might require specific placement cleaning cartridges. See the manufacturer's documentation for directions.
- Enable the SAM-QFS auto-cleaning feature:

1. Disable the media changer's cleaning feature, according to the manufacturer's documentation.
2. Edit the `defaults.conf` file to add the following line:

```
tapeclean = all autoclean on logsense on
```

   The `logsense` option prevents a drive from using expired cleaning media. If you prefer to use only sense data for determining the status of cleaning media, use the following line:

```
tapeclean = all autoclean on logsense off
```

Note -

When using the auto-cleaning feature with a library with more than two drives, use at least two cleaning cartridges for each catalog. If not enough cleaning cartridges are available, any drive that requires cleaning is put into a `DOWN` state.

How To Clean a Tape Drive Manually

When automatic cleaning is not available and the system uses barcodes, you can request that a drive be cleaned at any time.

- Use the following command:

```
cleandrive <eq>
```

  For eq, specify the equipment number of the automated library as defined in the `mcf` file.
  This drive is loaded with the cleaning cartridge.

How To Clear Media Errors

Note -
Removing the error flag can cause problems. If you are not certain of what caused the error and that the flag can be removed safely, contact Sun Technical Support and do not use this procedure.

When a hardware or software error is encountered on a cartridge, the SAM-QFS system sets the `media error` flag in the VSN catalog. The `media error` flag is displayed in the `samu`(1M) utility's `v` display and in SAM-QFS Manager.
You can clear the error to reset the flag and you can then attempt to use the cartridge.

1. To clear the `media error` flag on a cartridge:

```
chmed -E <media-type>.<vsn>
```

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

2. Update the library catalog with the space remaining information.

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

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

For more information, see the `auditslot`(1M) man page.
You can also perform this task by using the `samu`(1M) utility's `:audit` command or SAM-QFS Manager.

How To Remove a Stuck Cartridge From a Drive

1. Use the `samcmd off` command to turn off the drive in the automated library.

```
samcmd off <eq>
```

   For eq, specify the equipment number of the drive as defined in the `mcf` file.
   You can also perform this step by using `samu`(1M) or SAM-QFS Manager.

2. Use the `samcmd off` command to turn off the automated library.

```
samcmd off <eq>
```

   For eq, specify the equipment number of the library as defined in the `mcf` file.
   You can also perform this step by using `samu`(1M) or SAM-QFS Manager.

3. Physically remove the cartridge from the drive.
   Make sure you do not damage either the cartridge or the drive.
4. Use the `samcmd on` command to turn on the automated library and the drive.
   Issue this command once for the drive and once for the library.

```
samcmd on <eq>
```

For eq, specify the equipment number of the library and then of the drive as defined in the `mcf` file.
If the automated library performs an audit when it is turned on, you are done.

5. Follow these steps if the automated library does not perform an audit:
   a. Put the cartridge back into its storage slot.
   b. Use the `chmed`(1M) command to adjust the library catalog to set the occupied flag for the damaged tape.

   ```
   chmed +o <eq>:<slot>
   ```

Argument	Meaning
eq	The equipment number of the automated library or drive as defined in the `mcf` file.
slot	The number of the storage slot in the library as recognized in the library catalog. This argument is not applicable for manually loaded drives.

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

Catalog Operations, Importing Cartridges, and Exporting Cartridges

The physical addition (import) of cartridges to and removal (export) of cartridges from an automated library enables you to perform several functions, including the following:

- Replace cartridges.
- Relocate cartridges to off-site storage to use later for disaster recovery purposes. You can use the `-I` option on the `chmed`(1M) command to specify additional information such as the storage location of the cartridge.

When you import and export cartridges, you also update the library catalog.

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

The SAM-QFS environment treats catalogs differently depending on how the automated library is attached to the server, as follows:

- If the automated library is direct attached, the library catalog is a one-to-one mapping between library catalog entries and physical slots in the automated library. The first entry in the library catalog is for the first slot in the automated library. When a cartridge is needed, the system consults the library catalog to determine which slot contains the VSN, and it issues a command to load the cartridge from that slot into a drive.

> Note -
> SAM-QFS does not support mixed media in direct attached libraries. If the library is partitioned, each partition must contain only one media type.

- If the automated library is network attached, the library catalog is not a direct mapping to the slots. It is a list of the VSNs known to be present in the automated library. When a cartridge is requested, the system sends a request to the vendor's software to load the VSN into a drive. The vendor's software locates the VSN's storage slot.

Each automated library handles cartridge import and export differently due to system characteristics and the vendor-supplied software. For example, on the ACL 4/52 library, you need to issue a `move` command to move cartridges into the import/export unit before exporting cartridges from the automated library.

Network attached automated libraries import and export cartridges using their own utilities, so the `samimport`(1M) and `samexport`(1M) commands update only the library catalog entries used by the SAM-QFS systems. If you have a network attached library, see Managing Vendor-Specific Libraries for information about importing and exporting cartridges.

Tracking Exported Media - The Historian

The SAM-QFS historian keeps track of cartridges exported from an automated library or a manually mounted device. The historian acts like a virtual library, but it has no defined hardware devices. It has the following similarities to an automated library:

- Is configured in the mcf file
  You can configure the historian in the mcf file by using a device type of hy. If you do not configure the historian in the mcf file, it is created as follows:

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

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

- Has a catalog that records entries for all cartridges associated with it
  The historian library catalog is initialized with 32 entries when the historian first starts. Make sure that the catalog resides on a file system large enough to hold the entire catalog. Your site might want to track existing SAM-QFS cartridges that have been exported from the library. In this case, you need to build a historian catalog from the existing cartridges as described in the build_cat(1M) man page.
  The following two configuration directives in the defaults.conf file affect the behavior of the historian:
  - The exported_media = unavailable directive flags any cartridges exported from an automated library as unavailable to the historian. Requests for these cartridges generate EIO errors.
  - The attended = no directive declares to the historian that no operator is available to handle load requests. Requests to load cartridges that are not already loaded generate EIO errors.
    For more configuration information, see the historian(7) and defaults.conf(4) man pages.
- Can import and export cartridges
  Importing and exporting practices differ from manufacturer to manufacturer. See Managing Vendor-Specific Libraries to determine if specialized procedures are recommended for your equipment.
- Is displayed in SAM-QFS Manager as another automated library

About Importing and Exporting From an Automated Library

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

How To Import a Cartridge From a Library With a Mailbox

1. Open the mailbox using the manufacturer's suggested operation, usually by a button near the mailbox. Sometimes the mailbox is a one-slot mailbox referred to as a mail slot in the vendor's documentation.
2. Manually place the cartridge in the mailbox.
3. Close the mailbox.
4. Use the samimport(1M) command to import the cartridge.

   ```
   samimport <eq>
   ```

   For eq, specify the equipment number of the library as defined in the mcf file.
   The system moves the cartridge from the mailbox to a storage slot and updates the library catalog for the cartridge.
   You can also perform this task by using samu(1M) or SAM-QFS Manager.

How To Export a Cartridge From a Library With a Mailbox

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

   - ```
     samexport <eq>:<slot>
     ```
   - ```
     samexport <media-type>.<vsn>
     ```

Argument	Meaning
eq	The equipment number of the automated library as defined in the mcf file.

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

You can also perform this step by using `samu(1M)` or SAM-QFS Manager.

1. Open the mailbox or mail slot using the manufacturer's instructions, usually by pressing a button near the mailbox.

How To Import a Cartridge From a Library Without a Mailbox

1. Use the following command:

```
samcmd unload <eq>
```

For eq, specify the equipment number of the library as defined in the `mcf` file.

2. Wait until the system completes its current task, sets the status to `off`, and transfers the current active catalog to the historian.
3. Unlock and open the door to the automated library.
4. Load cartridges into the available slots.
5. Close and lock the door to the automated library.

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

How To Export a Cartridge From a Library Without a Mailbox

1. Use the following command:

```
samcmd unload <eq>
```

For eq, specify the equipment number of the library being addressed as defined in the `mcf` file.

2. Wait until the system completes its current task, sets the status to `off`, and transfers the current active catalog to the historian.
3. Unlock and open the door to the automated library.
4. Remove the cartridges from their respective slots.
5. Close and lock the door to the automated library.

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

How To Enable Load Notification

The SAM-QFS software requests cartridges to be loaded regularly to satisfy archiving and staging needs. If the request is for a cartridge that resides inside a library, the request is handled automatically. If the request is for a cartridge that resides outside the library, operator action is required. If enabled, the `load_notify.sh` script sends email when a cartridge needs to be obtained from outside the library.

1. Become superuser.
2. Copy the load notification script from its installed location to its operable location. For example:

```
# cp /opt/SUNWsamfs/examples/load_notify.sh /etc/opt/SUNWsamfs/scripts/load_notify.sh
```

3. Use `more(1)` or another command to examine the `defaults.conf` file. Make sure that the following default directives are in the file and have not been changed.

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

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

Using Drives With Encryption Capability

If you are archiving files to drives with encryption capability, plan your archive operations according to the following special considerations:

- Do not mix non-encrypted and encryption-capable drives in a library.
- After a drive has encryption enabled, it cannot be disabled.
- Do not mix encrypted and non-encrypted files on a tape.
- An encrypted drive cannot append to a tape that contains non-encrypted data.
- An encryption-enabled drive can read non-encrypted data.

Manually Loaded Drive Operations

This section describes operations that differ if you have a manually loaded, stand-alone drive rather than an automated library. Each manually loaded drive has its own one-slot library catalog.

How To Load a Cartridge

- To load a cartridge into a manually loaded device, place the cartridge in the drive according to the manufacturer's instructions.
  The SAM-QFS system recognizes that the cartridge is loaded, reads the label, and updates the one-slot catalog. No further action is necessary.

How To Unload a Cartridge

- Use `samcmd idle` command to idle the drive.

  ```
  samcmd idle <eq>
  ```

  For eq, specify the equipment number of the drive as defined in the `mcf` file.

This command ensures that no archive or stage processes are active. The drive switches from `idle` to `off` when all I/O activity is complete, and the tape ejects.
If the cartridge is a tape, the tape rewinds and is ready to be removed. An optical disk ejects automatically. See the manufacturer's instructions for removing the specific cartridge.
You can also perform this task by using `samu`(1M) or SAM-QFS Manager.

How To View a Library Catalog

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

  ```
  :v <eq>
  ```

  For eq, specify the equipment number of the library as defined in the `mcf` file.

Managing Vendor-Specific Libraries

Contents

Managing Vendor-Specific Libraries

You can include libraries from many different manufacturers in a Sun Storage Archive Manager (SAM-QFS) environment. For most libraries, use the operational procedures described in Managing Automated Libraries and Manually Loaded Drives. The following libraries, however, have vendor-specific operational procedures:

- ADIC/Grau Automated Libraries
- IBM 3584 UltraScalable Tape Libraries
- IBM 3494 Libraries
- Sony Direct Attached 8400 PetaSite Automated Libraries
- Sony Network Attached Automated Libraries
- StorageTek ACSLS-Attached Automated Libraries

Note -
Consult your Oracle sales representative or your authorized service provider for information about library model numbers, firmware levels, and other compatibility information.

ADIC/Grau Automated Libraries

Note -
ADIC/Grau network attached libraries are not supported by SAM-QFS software on an x64 hardware platform.

The following sections describe aspects of this library's operations that differ from the procedures described in Managing Automated Libraries and Manually Loaded Drives.

Because you use vendor-supplied utilities to physically add and remove cartridges in the ADIC/Grau automated library, the SAM-QFS interfaces (`samimport`(1M), `samexport`(1M), and SAM-QFS Manager) affect only the library catalog.

How To Import a Cartridge

1. Use ADIC/Grau commands to physically move the cartridge into the library.
2. Use the `samimport`(1M) command to update the library catalog.

```
samimport -v <volser> <eq>
```

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

How To Export a Cartridge

1. Use the samexport(1M) command to remove the entry from the library catalog.

```
samexport <eq>:<slot>
samexport <media-type>.<vsn>
```

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

   The samexport(1M) command updates the library catalog as each VSN is exported. It also moves the library catalog entry for each VSN from the library catalog to the historian.

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

IBM 3584 UltraScalable Tape Libraries

> Note -
> IBM 3584 UltraScalable libraries are not supported by the SAM-QFS software on an x64 hardware platform.

The following sections describe aspects of this library's operations that differ from the procedures described in Managing Automated Libraries and Manually Loaded Drives.

Importing Cartridges

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

Cleaning Drives

Disable automatic cleaning and enable hosted cleaning. This process is described in the IBM 3584 UltraScalable Tape Library Planning and Operator Guide, IBM publication GA32-0408-01. This process is also described in the ibm3584(7) man page.

Partitioning

This library accommodates several tape drives. If you use multiple drives, you can divide this one physical library into two, three, or four logical libraries. If you do divide your library into two or more logical libraries, be sure these logical libraries are operating properly before you add the IBM 3584 library to the SAM-QFS environment.

When a cartridge is exported from a partitioned library, only the logical library from which it was exported can get access to that drawer slot. If the cartridge is removed and reinserted manually, any logical library can get access to the drawer slot.

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

How To Remove a Cartridge From a Logical Library

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

IBM 3494 Libraries

> Note -
> IBM 3494 network attached libraries are not supported by the SAM-QFS software on an x64 hardware platform.

The following sections describe aspects of this library's operations that differ from the procedures described in Managing Automated Libraries and Manually Loaded Drives.

How To Import a Cartridge

1. Place the new media into the I/O slots.
2. Close the door.
   The library locks the door and moves the media into the storage area. You can import only 100 volumes at one time.
   a. If the library is configured with `access=private`, the library informs the daemon as the media is moved and the media is added to the catalog.
   b. If the library is configured with `access=shared`, issue the `samimport`(1M) command to add the media to the catalog.

How To Export a Cartridge

1. Use the `export`(1M) command to export cartridges.
   This command moves the media to the I/O area and turns on the output mode light on the operator panel.
2. Physically remove the media from the I/O area.

Sony Direct Attached 8400 PetaSite Automated Libraries

> Note -
> Sony 8400 PetaSite libraries are not supported by the SAM-QFS software on an x64 hardware platform.

The Sony 8400 PetaSite Series automated library is different from other Sony models because it has an eight-slot import and export mailbox (slots 400-407). Because the mailbox slots can be used as storage slots, the SAM-QFS library catalog keeps track of the mailbox slots. This automated library uses a barcode reader.

> Note -
> The information in this section applies only to Sony direct attached 8400 PetaSite automated libraries. This information does not pertain to the Sony direct attached B9 and B35 automated libraries, nor does it pertain to the Sony Network Attached Automated Libraries .

How To Import Tapes

1. Open the door of the automated library by pushing the open/close button on the front panel of the automated library.
2. Load the cartridges into the mailbox slots.
3. Push the open/close button on the front panel of the automated library
4. Manually close the door to the mailbox.
   The automated library checks the mailbox slots for the cartridge barcodes. If the library detects a problem with a barcode, both the `in` and `out` lights flash for that slot.
5. Use the `samimport`(1M) command to enable the SAM-QFS system to recognize the imported cartridges.

```
samimport <eq>
```

   For eq, specify the equipment ordinal of the device being addressed as defined in the `mcf` file.
   You can also perform this step by using SAM-QFS Manager. For more information, see SAM-QFS Manager online help.

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

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

```
move <source-slot> <destination-slot> <eq>
```

Argument	Meaning
source-slot	The number of the slot in which the cartridge currently resides.
destination-slot	The number of the slot into which the cartridge should be moved.
eq	The equipment ordinal of the device being addressed as defined in the mcf file.

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

```
samexport <eq>
```

For eq, specify the equipment ordinal of the device being addressed as defined in the mcf file.
You can also perform this step by using SAM-QFS Manager. For more information, see SAM-QFS Manager online help.

How To Export a Tape Using Mailbox Slots As Storage Slots

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

```
samexport <eq>
```

For eq, specify the equipment ordinal of the device being addressed as defined in the mcf file.
You can also perform this step by using SAM-QFS Manager. For more information, see SAM-QFS Manager online help.

How To Move a Cartridge to a Different Slot

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

```
move <eq>:<source-slot> <destination-slot>
```

Argument	Meaning
eq	The equipment ordinal of the device being addressed as defined in the mcf file.
source-slot	The number of the slot in which the cartridge currently resides.
destination-slot	The number of the slot into which the cartridge should be moved.

You can also perform this step by using SAM-QFS Manager. For more information, see SAM-QFS Manager online help.

Sony Network Attached Automated Libraries

Note -
Sony network attached libraries are not supported by the SAM-QFS software on an x64 hardware platform.

The following sections describe aspects of this library's operations that differ from the procedures described in Managing Automated Libraries and Manually Loaded Drives.

Because you use vendor-supplied utilities to physically add and remove cartridges, the SAM-QFS interfaces (samimport, samexport, and SAM-QFS Manager) affect only the library catalog.

How To Import a Cartridge

1. Use Sony commands to physically move the cartridge into the library.
2. Use the samimport(1M) command to update the library catalog.

```
samimport -v "<volser>" <eq>
```

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

How To Export a Cartridge

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

```
samexport <eq>:<slot>
samexport <media-type>.<vsn>
```

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

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

StorageTek ACSLS-Attached Automated Libraries

The following sections describe aspects of this library's operations that differ from the procedures described in Managing Automated Libraries and Manually Loaded Drives.

Some StorageTek automated libraries, such as the StorageTek 9730 library use a mail slot to import and export only one cartridge at a time. A mailbox is an area used for putting cartridges into and removing cartridges from the automated library. Examples of StorageTek automated libraries that have a mailbox are the StorageTek 9714 and the StorageTek 9710 libraries.
In StorageTek documentation, the mailbox and mailslot are both referred to as the cartridge access port (CAP).

When importing and exporting cartridges from any ACSLS-attached automated library, you must keep the ACSLS inventory and the SAM-QFS catalog in agreement.

- When importing cartridges, the samimport(1M) command does not insert cartridges physically into the automated library. You must also issue ACSLS commands to complete the operation.
- When exporting cartridges, issue the samexport(1M) command with its -f option to direct the SAM-QFS system to put the cartridge in the CAP and to update the catalog. Without the -f option, the cartridge is not in the CAP so you must then use ACSLS commands to complete the operation.

You can also perform the import and export procedures by using samu(1M) or SAM-QFS Manager. For more information, see SAM-QFS Manager online help.

How To Import Tapes

- Use the `samimport`(1M) command in the following format:

```
samimport -v <vsn eq>
```

Argument	Meaning
vsn	The volume serial name (VSN) assigned to the volume.
eq	The equipment ordinal of the device being addressed as defined in the `mcf` file.

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

How To Export Tapes Using a Mailbox

You can export tape cartridges by slot or by VSN. Export the tape using one of the following formats:

- ```
  samexport [-f] <eq>:<slot>
  ```
- ```
  samexport [-f] <media-type>.<vsn>
  ```

Argument	Meaning
`-f`	Specification for the SAM-QFS system to put the volume in the cartridge access port (CAP) and to update the catalog accordingly.
eq	The equipment ordinal of the device being addressed as defined in the `mcf` file.
slot	The number of a storage slot in an automated library as recognized in the library catalog.
media-type	The media type. For a list of valid media types, see the `mcf`(4) man page.
vsn	The volume serial name (VSN) assigned to the volume.

More About SAM-QFS

Contents

More About SAM-QFS

The Sun Storage Archive Manager (SAM-QFS) software provides a configurable file system with storage, archive management, and retrieval capabilities. The SAM-QFS software archives files by copying the files from online disk cache to archive media. The archive media can consist of disk slices in another file system or it can consist of removable tape or magneto-optical cartridges in automated or manually loaded storage devices. The SAM-QFS software automatically maintains online disk space at site-specified usage thresholds. It releases disk space associated with archived file data and restores the files to online disk when they are needed.

Components of SAM-QFS

The SAM-QFS software consists of the following components which define the archiving lifecycle:

- Archiving

- Releasing
- Staging
- Recycling

Archiving

The archiver automatically copies online disk cache files to archive media. The archive media can consist of either online disk files or removable media cartridges. The archiver requires you to configure the `archiver.cmd` file to define what to archive. You can configure the archiver to create up to four archive copies on a variety of archive media. If a file is segmented, each segment is treated as a file and is archived separately. The archiving process is initiated after disk-based files match a site-definable set of selection criteria.

For more information about the archiver, see About Archiving.

Releasing

The releaser automatically maintains the file system's online disk cache at site-specified percentage usage thresholds by freeing disk blocks occupied by eligible archived files.

Releasing is the process of freeing primary (disk) storage that is used by an archived file's data. Two threshold values, the high-water mark and the low-water mark, both expressed as a percentage of total disk space, are used to manage free space in the online disk cache. When online disk consumption exceeds the high-water mark, the system automatically begins releasing the disk space occupied by eligible archived files. Disk space occupied by archived file data is released until the low-water mark is reached.

Files are selected for release depending on their size and age. The first portion of a file can be retained on disk for speedy access and for masking staging delays. If a file has been archived in segments, portions of the file can be released individually.

For more information about the releaser, see About Releasing.

Staging

The stager restores file data to the disk cache. When a user or process requests file data that has been released from disk cache, the stager automatically copies the file data back to the online disk cache.

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

The SAM-QFS software processes stage request errors automatically. If a stage error is returned, the system attempts to find the next available archive copy of the file. Stage errors that can be automatically processed include media errors, unavailability of media, unavailability of an automated library, and others.

For more information about staging, see About Staging.

Recycling

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

As users modify files, archive copies associated with the old versions of these files are considered to be expired on their archive media. Because these copies are no longer needed, they can be purged from the system. The recycler identifies the archive volumes with the largest proportions of expired archive copies and preserves the unexpired copies by moving them to separate volumes. The recycling process is transparent to end users.

If a removable media volume contains only expired copies, you can take one of the following actions:

- Relabel the volume for immediate reuse.
- Export the volume to offsite storage as a historical record of file changes. You can use standard UNIX utilities to restore previous versions of files from expired archive copies.

For more information about recycling, see About Recycling.

Supported Storage Devices

The SAM-QFS environment supports a wide variety of tape storage and magneto-optical devices. The automated libraries that SAM-QFS supports can be divided into the following groups, depending on how they are attached to the environment:

- A direct attachment. A direct attached library is connected directly to the host system using a Small Computer System Interface (SCSI). This connection can be either a direct connection or a Fibre Channel connection. For example, a direct attachment is used for Sun

StorageTek libraries. The SAM-QFS system controls these libraries directly using the SCSI standard for automated libraries.
- A network attachment. The SAM-QFS software can be configured as a client of the library's host system. The network attached libraries include some of the StorageTek, ADIC/Grau, IBM, and Sony libraries. These libraries use a software package supplied by the vendor. In these cases, the SAM-QFS software interfaces with the vendor software, using a daemo designed for the automated library.

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

When possible, the system uses the standard Solaris disk and tape device drivers. For devices not directly supported in the Solaris OS, such as certain library and optical disk devices, the SAM-QFS software packages include special device drivers.

For a list of supported storage devices, contact your Oracle sales representative or your authorized service provider (ASP). For information about configuring storage devices, see Configuring Storage Devices for Archiving.

SAM-Remote Software

Sun SAM-Remote software is a client/server implementation that enables libraries and other removable media devices to be shared between SAM-QFS host systems. Sun SAM-Remote software enables you to configure multiple storage clients that archive and stage files from a centralized tape library or magneto-optical library. For example, if you have host systems on a network that spans a large geographical area, files created in one city can be archived to cartridges in a library located miles away.

For more information see Using the Sun SAM-Remote Software.

Populating the Catalog

Contents

Populating the Catalog

After you mount a file system, the SAM-QFS software creates catalogs for each automated library configured in the `mcf` file. However, if you have a network attached automated library, you need to populate the library's catalog. The appropriate method depends on the number of volumes you include in the catalog.

How to Populate an Automated Library With Many Volumes

Use this procedure for ADIC/Grau, Sony network attached, StorageTek ACSLS-Attached, and IBM 3494 automated libraries.

Note the following when creating the input file:

- The file has four fields in each row. Each row identifies a volume. For each volume, specify the slot number, the VSN, the bar code, and the media type.

  > Note -
  > The slot position of a tape in a network attached automated library has no relationship to the slot number of the volume in a Sun Storage Archive Manager (SAM-QFS) library catalog.

- Use a space character or a tab character to separate the fields in this file.
- If a VSN contains one or more space characters, enclose the VSN name in quotation marks (" ").

1. Create an input file that contains the slot number, the volume's VSN, the barcode number, and the media type.
   The following example shows the sample file `input_vsns`.

```
0 TAPE01  "TAPE 01" lt
1 TAPE02  TAPE02  lt
2 TAPE03  TAPE03  lt
```

2. Use the `build_cat`(1M) command to create the catalog.

```
build_cat <input-file> <catalog-file>
```

Argument	Content
input-file	The name of an input file. Typically, this is a file containing a list of VSNs.
catalog-file	The full path to the library catalog. By default, the Sun Storage Archive Manager software creates a catalog and writes it to `/var/opt/SUNWsamfs/catalog`/family-set-name, where family-set-name is derived from the `mcf` file entry for this automated library. Alternatively, if you have specified a catalog name in the Additional Parameters field of the `mcf` file, use that catalog file name for catalog-file.

For example, you might specify the following `build_cat`(1M) command:

```
# build_cat input_vsns /var/opt/SUNWsamfs/catalog/grau50
```

How to Populate an Automated Library With a Small Number of Volumes

Use this procedure for ADIC/Grau, Sony network attached, StorageTek ACSLS-Attached, and IBM 3494 automated libraries.

Perform this procedure for each cartridge that you want to include in the catalog. The cartridge must be physically present in the automated library. If the cartridge is not present, the entry is recorded in the historian, which keeps track of cartridges exported from an automated library or a manually mounted device and you see an error message, as described in Example 3 - Message Generated After a `samimport`(1M) Command. For more information about the historian, see Tracking Exported Media-The Historian.

- Use the `samimport`(1M) command to import catalog entries into the default catalog.

```
samimport -v <VSN> <eq>
```

Argument	Content
VSN	Specify the VSN identifier for a volume. If a VSN name contains one or more space characters, enclose the VSN name in quotation marks (" ").
eq	Specify the Equipment Ordinal as specified for the device in the `mcf` file.

For example:

```
# samimport -v TAPE01 50
```

How to Populate an IBM 3494 Automated Library

Use this procedure only if you are using an IBM 3494 library as one physical library, that is, if `access=private` is specified in the `mcf` file. Do not use this procedure if you divided the library into multiple logical libraries.

Note -
If you have an IBM 3494 library that is divided into multiple logical libraries, that is, `access=shared` is specified in the IBM 3494 parameters file, use one of the previous methods to populate the catalog: How to Populate an Automated Library With Many Volumes or How to Populate an Automated Library With a Small Number of Volumes.

- Insert the media cartridge into the mail slot.
  The library automatically builds a catalog that includes the media cartridge.

How to Populate a StorageTek ACSLS-Attached Library Quickly

This procedure is an alternative, faster method of populating a library catalog than the methods described in How to Populate an Automated Library With Many Volumes or How to Populate an Automated Library With a Small Number of Volumes.

- Use the `samimport`(1M) command with its `-c` and `-s` options to import from a pool of VSNs.
  For more information, see the `samimport`(1M) man page.

StorageTek ACSLS-Attached Automated Libraries: Common Problems and Error Messages

If errors exist in the configuration files for a StorageTek ACSLS-attached automated library, the system generates several error messages. The following examples show common problems and the messages that the system generates.

Example 1 - Errors From a StorageTek ACSLS Parameters File

```
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 your StorageTek parameters file for syntax errors.
Each line must begin with a keyword or a comment.
For more information about the StorageTek parameters file, see the `stk`(7) man page.

Example 2 - Error Messages from a StorageTek ACSLS Library

The following example shows drives that are frozen in the initializing state.

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

The following example shows the `samu`(1M) utility's `:r` display.

```
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 frozen in an initializing state or that do not initialize indicate a configuration error.

- Verify that ACSLS software is running.
- Verify the host name.
- Determine whether you can you reach the host using the `ping`(1M) command.
- Check the `portnum` specification in the StorageTek parameters file. In ACSLS 5.3, for example, the default port number, 50004, is used for a different application. Try a higher port number, such as 50014.

Example 3 - Message Generated After a **`samimport`**(1M) Command

Example 3 shows messages generated after the samimport(1M) command was used to import a VSN to the library catalog, but the VSN was not in the StorageTek automated library. The cartridge must be present in the ACSLS-managed automated library before the samimport(1M) command can be successful.

```
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
```

> Note -
> The sam-stkd daemon uses the ssi.sh script to ensure that a copy of the SSI daemon, ssi_so, is running. If ssi_so exits, the daemon starts another. If your site has its own version of ssi.sh, modify it to wait for a SIGTERM signal and then exit. The daemon sends a SIGTERM signal to stop the process. The /opt/SUNWsamfs/examples/ssi.sh file contains an example ssi.sh script. The system copies the ssi.sh script to /etc/opt/SUNWsamfs/scripts/ssi.sh during installation if one does not already exist.

Using the Sun SAM-Remote Software

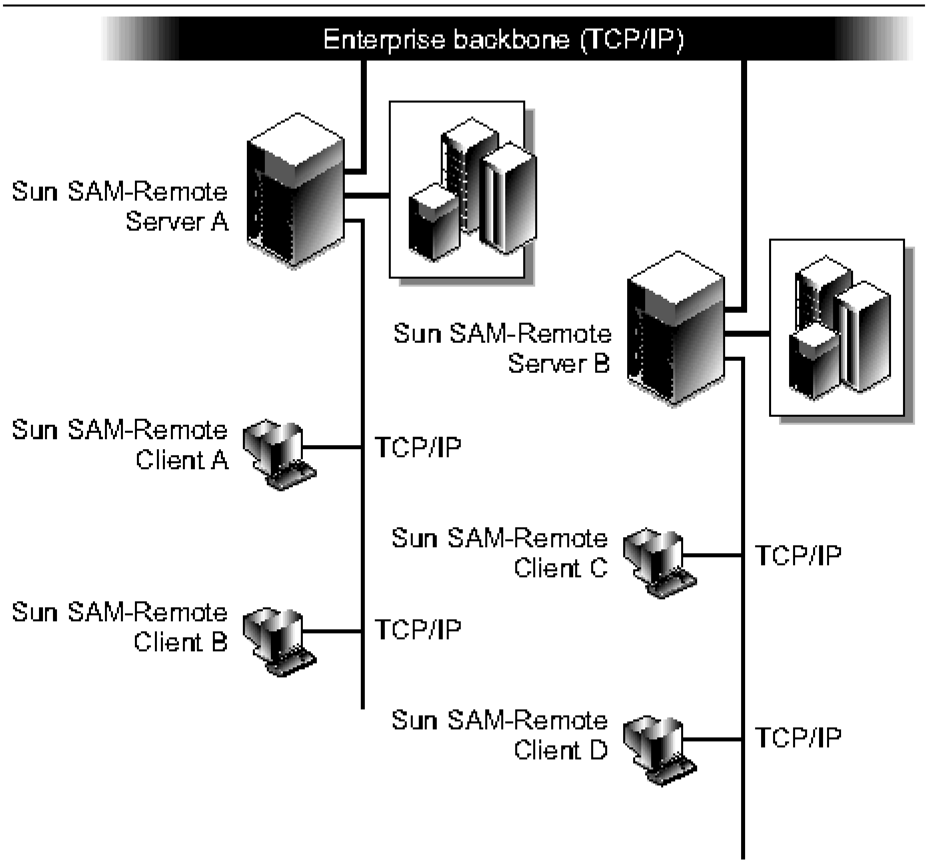
Contents

Using the Sun SAM-Remote Software

The Sun SAM-Remote client and the Sun SAM-Remote server form an implementation that allows libraries and other removable media devices to be shared between SAM-QFS host systems. Use the SAM-Remote software to configure multiple storage clients that archive and stage files from a centralized tape library or magneto-optical library. For example, if you have host systems on a network that spans a large geographical area, files created in one city can be archived to cartridges in a library located miles away.

Sun SAM-Remote Software Overview

Figure - Two Sun SAM-Remote Servers, Each With Two Clients



Sun SAM-Remote software provides the following advantages:

- Enables you to configure remote sharing of an expensive removable media resource, such as a library, between two or more Sun SAM-Remote clients.
- Enables clients to migrate data to a server.
- Enables multiple SAM-QFS servers to be hosts to one another. In a Sun SAM-Remote environment, the server is the host system that is configured with an equipment type of `ss` in the `mcf` file.

You can configure the Sun SAM-Remote server and clients to provide multiple archive copies between two or more Sun Solaris host systems. For example, you can configure two Solaris systems running SAM-QFS software as both Sun SAM-Remote servers and Sun SAM-Remote clients to each other. Benefits of this configuration include the ability to create local copies for each server with an additional archive copy of data on the other server. File systems can be shared between servers using standard NFS. In the event of a loss of access to the local library, Sun SAM-Remote software would automatically retrieve file data from the archive copy. Users of both servers would have uninterrupted access to their data even if their primary storage library were unavailable.

System Requirements

Before attempting to configure a Sun SAM-Remote environment, make sure that your environment includes the following software and hardware:

- SPARC® or x64 systems with licensed, installed, and operable SAM-QFS archive management software packages.
- Host systems with identical SAM-QFS software revision levels and identical patch collections installed. If some host systems have to be upgraded, see Upgrading QFS and SAM-QFS.
- One or more host systems to act as the Sun SAM-Remote server with at least one SAM-QFS file system.
- A network connection running a TCP/IP connection between the clients and the server.

Software Limitations

The SAM-QFS software treats cartridges in a remote library no differently than it treats cartridges in a local library. The following information, however, indicates the limits of Sun SAM-Remote software:

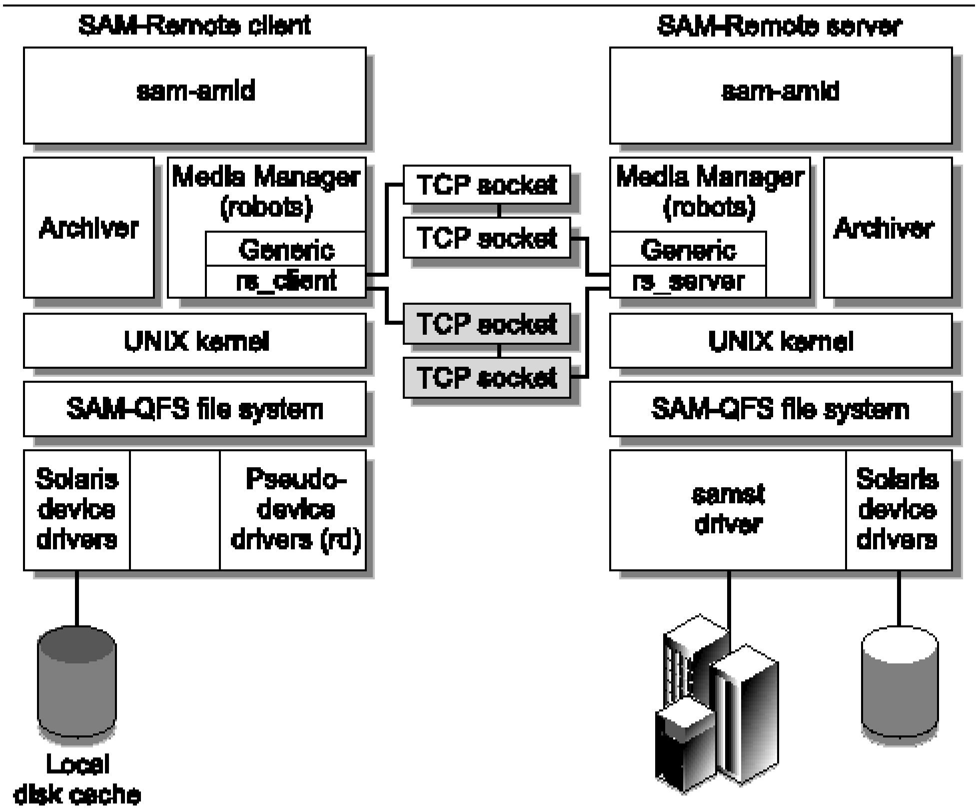
- You can recycle media using Sun SAM-Remote, but you attempt this activity only after thoroughly testing your environment. For more information, see Recycling With the Sun SAM-Remote Software.
- Only one daemon on a Sun SAM-Remote client can communicate to the Sun SAM-Remote server.
- SAM-QFS software, and therefore Sun SAM-Remote, cannot operate on Sun QFS clients in a shared Sun QFS file system. When running on a server that is a metadata server for some Sun QFS file systems and a client for other Sun QFS file systems, SAM-QFS software and

Sun SAM-Remote operate only on the file systems for which that server is a metadata server.

Client and Server Interactions

Sun SAM-Remote clients interact with the Sun SAM-Remote server using a TCP/IP connection. The network between Sun SAM-Remote clients can be any network type supported by the Solaris OS, such as Ethernet, Fast Ethernet, or Fibre Channel.

Figure - Sun SAM-Remote Server and Client Interactions



Sun SAM-Remote Server Overview

The Sun SAM-Remote server consists of a full-capability SAM-QFS storage management host and a Sun SAM-Remote server daemon that defines libraries to be shared among the clients. At least one SAM-QFS file system must be configured on the Sun SAM-Remote server.

You define a host system as a Sun SAM-Remote server by adding a line in the server system's `/etc/opt/SUNWsamfs/mcf` file with an equipment type of `ss`. You must provide a unique family set name for each server. Up to ten clients can be configured per daemon. To configure more than ten clients, add an additional remote server entry in the `mcf` file for each ten clients that you want to configure. For more information about the server daemon, see the `sam-remote`(7) man page.

Sun SAM-Remote Client Overview

The Sun SAM-Remote client is a SAM-QFS host system that establishes a Sun SAM-Remote client daemon containing a number of pseudo-devices.

You define a host system as a Sun SAM-Remote client by adding a line in the client system's `/etc/opt/SUNWsamfs/mcf` file with an equipment type of `sc`. For more information about the client daemon, see the `sam-remote`(7) man page.

A pseudo-device defines a network connection to an actual removable media device on the Sun SAM-Remote server. Pseudo-devices have an equipment type of `rd`, which is an abbreviation for remote device. You define the pseudo-devices in the Sun SAM-Remote client's `/etc/opt/SUNWsamfs/mcf` file. The Sun SAM-Remote daemon and pseudo-devices are associated with one particular server.

The Sun SAM-Remote daemon supports an unlimited number of pseudo-devices for each client. The actual number of pseudo-devices to be used by the client is configurable. When determining how many pseudo-devices should be configured per client, think of these devices as the number of simultaneous data transfers that can occur between the client and the server. As more pseudo-devices are defined, the possibility of increasing the total network traffic load increases. As the system administrator, determine the actual number of pseudo-devices needed for the system.

Interaction Between the Sun SAM-Remote Server and the Sun SAM-Remote Client

The Sun SAM-Remote server daemon, `sam-serverd`, listens for the clients on port 1000. You can configure a different port in the Sun Solaris `/etc/services` directory with a service name of `rmtsam`. When a Sun SAM-Remote client connects to the Sun SAM-Remote server, the `sam-serverd` daemon establishes a connection on another port and communicates this port number to that client, using the defined port. The socket size is passed to the client. The socket size is configurable and is described in more detail in Configuring the Sun SAM-Remote Software.

Library Catalogs

The Sun SAM-Remote library catalog is a subset of the catalog located on the Sun SAM-Remote server. The client catalog is updated in real time. The slots allotted to a Sun SAM-Remote client catalog are controlled only by the Sun SAM-Remote server.

After initialization, the system builds a client catalog and passes it to the Sun SAM-Remote client based on information from the Sun SAM-Remote server catalog file. After the connection between the host and client is established, media available to the client is flagged as available. If the connection between the client and server is lost, the media on the client side is flagged as unavailable. You can view the media availability through the `samu`(1M) v display. The information that appears in the `samu`(1M) v display on the client is a subset of what appears in the v display on the server. A good practice is to access the media catalog through the `samu`(1M) v display on Sun SAM-Remote server. For more information about the Sun SAM-Remote server client file, see Configuring the Sun SAM-Remote Software. For information about the `samu`(1M) operator utility, see Using the samu Operator Utility.

Changes to the catalog are passed between hosts as necessary. Any changes in the server catalog that involve a media type associated with a client are passed to the client, and the client catalog is updated.

Archiving

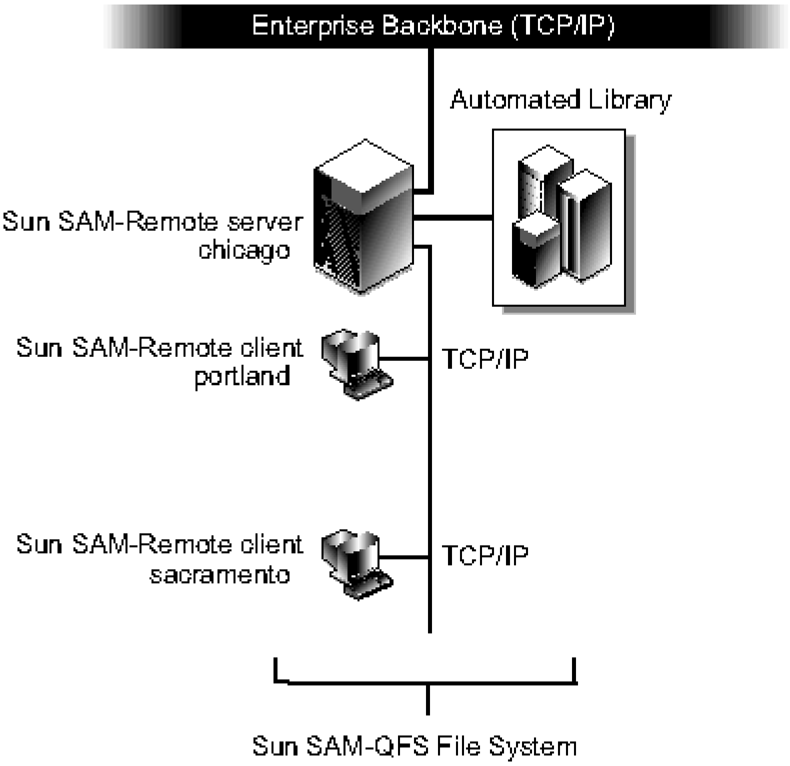
Sun SAM-Remote archive processing is the same as SAM-QFS archive processing. The Sun SAM-Remote client makes a mount request to be added to the server's mount request table. The client then waits for the server to respond with a message indicating that the media is mounted. Archiving begins when the media is available.

Configuring the Sun SAM-Remote Software

This section explains how to perform an initial configuration of the Sun SAM-Remote server and client software.

Figure - Example Sun SAM-Remote Configuration

The SAM-QFS file systems on `portland` and `sacramento` are clients of the Sun SAM-Remote server on `chicago`.



The following procedures explain how to configure the Sun SAM-Remote software on a Sun SAM-Remote server and on one or more Sun

SAM-Remote clients, using the example configuration.

- Perform the procedures in the order described.
- You must have superuser access to the server system on which the Sun Sun SAM-Remote software is to be installed.
- You must have superuser access to the client system or systems on which the Sun SAM-Remote software is to be installed.
- The same release and revision level of SAM-QFS software must be installed on all client and server hosts in the Sun SAM-Remote environment.
- The same patch collection must be installed on all client and server hosts in the SAM-Remote environment.

Step 1: Log In to the Potential Server and Client Hosts

1. Log in to the Sun SAM-Remote server as the superuser.
2. Log in to the Sun SAM-Remote clients as the superuser.

Step 2: Verify Client and Server Software

Follow these steps on each system to be configured as part of a Sun SAM-Remote environment.

1. Issue the `pkginfo`(1M) command with its `-l` option, and examine the output.

```
chicago#  pkginfo -l SUNWsamfsr
   PKGINST:  SUNWsamfsr
      NAME:  Sun SAM and Sun SAM-QFS software Solaris 10 (root)
  CATEGORY:  system
      ARCH:  sparc
   VERSION:  5.0.7,REV=5.10.2009.04.09
   BASEDIR:  /
    VENDOR:  Sun Microsystems, Inc.
      DESC:  Storage and Archive Manager File System
    PSTAMP:  wiffleball-mn20090409140022
  INSTDATE:  Jul 29 2009 11:00
   HOTLINE:  Please contact your local service provider
    STATUS:  completely installed
     FILES:      634 installed pathnames
                  23 shared pathnames
                  24 linked files
                  78 directories
                 199 executables
                   1 setuid/setgid executables
               74889 blocks used (approx)


chicago# pkginfo -l SUNWsamfsu
   PKGINST:  SUNWsamfsu
      NAME:  Sun SAM and Sun SAM-QFS software Solaris 10 (usr)
  CATEGORY:  system
      ARCH:  sparc
   VERSION:  5.0.7,REV=5.10.2009.04.09
   BASEDIR:  /
    VENDOR:  Sun Microsystems, Inc.
      DESC:  Storage and Archive Manager File System
    PSTAMP:  wiffleball-mn20090409140036
  INSTDATE:  Jul 29 2009 11:01
   HOTLINE:  Please contact your local service provider
    STATUS:  completely installed
     FILES:       57 installed pathnames
                  17 shared pathnames
                  20 directories
                  19 executables
                9597 blocks used (approx)


chicago#
```

The output shows that the server (`chicago`) is running software version 4U0.5. Any systems included in an environment with this server must also run version 4U0.5.

2. Issue the `showrev`(1M) command with its `-p` option, and examine the output.

```
chicago# showrev -p | grep SUNWsamfs

chicago#
```

The output shows that the server has no SAM-QFS patches installed. Any systems included in an environment with this server must also run version and patch level.
If you need to perform any software upgrades, see Upgrading QFS and SAM-QFS.

Step 3: Edit the **mcf** Files on Each Client

The mcf file defines a file system. This procedure adds the definition of the host as a Sun SAM-Remote client.

1. From the Sun SAM-Remote server, stop the SAM-QFS functions.
2. Issue the samcmd(1M) command to idle each removable media. For more information about the samcmd(1M) command, see the samcmd(1M) man page.

```
# samcmd idle <eq1>
# samcmd idle <eq2>
.
.
# samcmd idle <eqn>
```

   where eq specifies the equipment ordinal of the removable media drive, as defined in the mcf file.
   As an alternative, you can also use the samu(1M) operator utility to idle the drives. For information, see Using the samu Operator Utility.

   > Note -
   > All drives must be idle before you issue the next command, samd stop, so that the archiver, stager, and other processes can complete current tasks, and cartridges can be unloaded and put into their storage slots.

3. Issue the samd(1M) command with its stop option to stop the sam-amld daemon and its child processes.

```
# /opt/SUNWsamfs/sbin/samd stop
```

4. On each client, open the /etc/opt/SUNWsamfs/mcf file.
5. Define each system as a client of the Sun SAN-Remote server, following the example. The example code shows the client portland's mcf file after it is edited to make portland a client of the Sun SAM-Remote server chicago.
6. Open the mcf file of another client. In the example, the client sacramento's mcf file is edited.
7. Copy the last set of lines from the first client's mcf file to next client's mcf file. In the example, copy the last set of lines from portland's mcf file to sacramento's mcf file.
8. Save and close the mcf files.

Example: **mcf** Files on the Clients

```
# mcf file on portland
#
# Define a Sun QFS file system
#
# Equipment                                     Eq  Eq Family Dev Additional
# Identifier                                    Ord Ty Set    St  Parameters
# ==========                                    === == ====== == ==========
samfs1                                          10  ms samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE5F00048F2Dd0s6 11  md samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE23000B24C2d0s6 12  md samfs1 on


# Define Sun SAM-Remote Client portland to Sun SAM-Remote server chicago
#
/etc/opt/SUNWsamfs/rmt200 200 sc chicagoss on /var/opt/SUNWsamfs/catalog/tcat
/dev/samrd/rd0            201 rd chicagoss on
/dev/samrd/rd1            202 rd chicagoss on
```

The `mcf` entry on the client consists of a single-line entry for the Sun SAM-Remote client and a pseudo-device entry, indicated by the `rd` equipment type, for each device you want to configure. A pseudo-device defines a network connection to an actual device on the Sun SAM-Remote server. Each entry uses the following fields:

Equipment Identifier	The full path name of the client configuration file. You create the client configuration file in Step 4.
Eq Ord	The number that identifies this client system. Use this number to specify the eq parameter.
Eq Ty	A two-letter mnemonic that identifies the type of the client system.
Family set	The family set name of the daemon to be used on this server. A Sun SAM-Remote server can have one server daemon per client. Note - The family set name on the sam-remote client must match the family set name on the sam-remote server.
Dev St	Device state: on or `off`
Additional parameters	An optional field. In this example, it is the path to the catalog file.

Step 4: Create a Sun SAM-Remote Client Configuration File

A Sun SAM-Remote client's configuration file contains one entry: the name of the Sun SAM-Remote server.

1. On each client, create the file to be used as the Sun SAM-Remote client configuration file in the location that you specified in the `mcf` file. For example:

```
portland# vi /etc/opt/SUNWsamfs/rmt200
```

2. Type the name of the Sun SAM-Remote server.
The following example shows the client configuration file for the client `portland`, specifying that its Sun SAM-Remote server is the `chicago` system.

Example: Client Configuration File

```
portland# cat /etc/opt/SUNWsamfs/rmt200
chicago
```

Step 5: Edit the Server's **mcf** File

1. On the Sun SAM-Remote server, edit the `/etc/opt/SUNWsamfs/mcf` file.
2. Add the entries that define the SAM-QFS file systems and this system as the Sun SAM-Remote server. You must have at least one SAM-QFS file system. The following example shows the `mcf` file on `chicago`.

Example: **mcf** Files on the Server

```
# mcf file on Sun SAM-Remote server chicago:
# Define a SAM-QFS file system
#

# Equipment                                     Eq  Eq Family Dev Additional
# Identifier                                    Ord Ty Set    St  Parameters
# ==========                                    === == ====== == ==========
samfs1                                          10 ms samfs1 on
/dev/dsk/c6t600A0B80004850A600000F8048EF90ADd0s6 11 md samfs1 on
/dev/dsk/c6t600A0B800048505600000E9D48EF91EEd0s6 12 md samfs1 on


# Define a tape library that client portland can use:
/dev/samst/c4t500104F0009C2F6Fu0 100 sn rb100 on /var/opt/SUNWsamfs/catalog/rb100.cat
/dev/rmt/0cbn                    101 li rb100 on
/dev/rmt/1cbn                    102 li rb100 on


# Define Sun SAM-Remote server chicago
#
/etc/opt/SUNWsamfs/rmt200         50 ss chicagoss on
```

Note -
The family set name on the sam-remote server must match the family set name on the sam-remote client.

Step 6: Create the Server's Configuration File

The Sun SAM-Remote server configuration file defines the disk buffer characteristics and media to be used for each client. Ten clients can be configured per server daemon. To support more clients, configure another Sun SAM-Remote server daemon.

1. On the server, create the Sun SAM-Remote server configuration file.
2. Add definitions of each client, using the following format:

```
<client-name>
[ <parameter1> ]
    media
    <eq> <media-type> <regex>
     [<eq> <media-type> <regex> ]
     [. . .]
   endmedia
```

Field	Definition and Requirements
client-name	Identifies the system to be served by this invocation of the Sun SAM-Remote daemon. Use the network name, its IP address, or a fully-qualified domain name. The first character in client-name must be the first character in the line.
parameter (optional)	Defines an attribute of the client, using a keyword = value pair. For example, you can use the parameter to specify the network block size to be used by this client's socket in kilobytes. The format for this parameter: `net_blk_size=<size>` where size is an integer from 4≤ size≤ 64. The default is 4, which specifies 4096 bytes. The line containing the parameter must start with space or tab characters.

`media` and `endmedia`	Keywords that contain the media definitions. The definitions within these two keywords define the media archive volumes that a client can use. Note - These keywords are required and must be indented with space or tab characters.
eq media-type (regex)	Defines a media archive volume that this client can use. Enclose the regex data with parentheses. Because network-attached libraries have mixed media, specify each media type on a separate line. Note - Use the space or tab characters to indent the media definitions. * eq: Equipment Number as shown in the `mcf` file. * media-type: Two-character specific media type, such as `li`. For information about valid media types, see the `mcf`(4) man page but do not use the generic media type. * (regex): The volume serial names (VSN) of the cartridges to which the files are archived, expressed as an extended regular expression. For information about extended regular expressions, see the `egrep`(1) man page. For information about regular expressions, see the `regcomp`(3C) man page.

For example, the following is a valid media type definition:

```
media
   100 li (VSN1)
   100 li (VSN2)
endmedia
```

Note -
Do not allow the same physical media cartridges to be used by more than one client. Also, if the Sun SAM-Remote server has its own file system outside of the Sun SAM-Remote environment, it is not recommended that a cartridge be used by both the client and the server.

The following example shows the server configuration file, `/etc/opt/SUNWsamfs/rmt200` for the Sun SAM-Remote server `chicago`. This file defines clients `portland` and `sacramento`.

Example: Server Configuration File

```
# Sun SAM-Remote server config file /etc/opt/SUNWsamfs/rmt200
#
portland
   media
      100 li (100031|100032|100034|100035|100037|100038)
      100 li (200001|200002|200003|200004|200005|200006)
   endmedia
#
#
sacramento
   media
      100 li (300141|300142|300143|300145|300146|300147)
      100 li (400001|400002|400003|400005|400006|400007)
   endmedia
```

Step 7: Enable Archiving

1. Verify the `archiver.cmd` file on each client. Depending on your configuration, you might need to perform the following tasks:
   a. Make sure that the VSNs that are defined in the server configuration file are assigned to the correct archive sets.
   b. Remove the following directives if they apply to archive sets to be archived in the library connected to the Sun SAM-Remote

server:

- `-tapenonstop`
- `-offline_copy direct`

2. Issue the samd(1M) command with its start option to start the SAM-QFS processes on the server and on the clients. Enter the following command on the clients and the server:

```
server# samd start
```

3. Check the status of the Sun SAM-Remote connections.
   a. Issue the samu(1M) command on the server and the clients. For more information, see the samu(1M) man page or Using the samu Operator Utility.
   b. On each client, view the the samu(1M) utility's (s) display.
   The following example shows the status s display on the Sun SAM-Remote client portland. The device type sc identifies the Sun SAM-Remote client. The message after that line indicates that a connection with the server chicago has been established.

Example: **samu }}({{s**) Display for a Client

```
Device status samcmd      5.0.7 20:44:09 Jul 30 2009
samcmd on portland

ty     eq state   device_name                        fs status
sc    200 on      /etc/opt/SUNWsamfs/rmt200          200  ---------r
        Remote server 10.1.229.92 connected
rd    201 on      /dev/samrd/rd0                     200  ----------

rd    202 on      /dev/samrd/rd1                     200  ----------

hy    203 on      historian                          203  ----------
```

   c. On the server, view the the samu(1M) utility's (s) display.
   The following example shows the samu(1M) status s display on the Sun SAM-Remote server chicago. The device type ss identifies the Sun SAM-Remote server.

Example: **samu }}({{s**) Display for a Server

```
Device status samcmd      5.0.7 20:40:05 Jul 30 2009
samcmd on chicago

ty     eq state   device_name                        fs status
sk      1 on      /etc/opt/SUNWsamfs/SL500_SAM         1  m--------r
        running
li      2 on      /dev/rmt/0cbn                        1  ---------p
        empty
li      3 on      /dev/rmt/1cbn                        1  ---------p
        empty
ss     50 on      /etc/opt/SUNWsamfs/rmt200           50  -------o-r

hy     51 on      historian                           51  ----------
```

   d. On the server, view the the samu(1M) utility's (R) display.
   The following example shows the samu(1M) Sun SAM-Remote R display from the Sun SAM-Remote server chicago.

Example: **samu** (**R**) Display for a Server

```
Remote server eq: 50    addr: 00003858  samu      5.0.7 20:41:38 Jul 30 2009


message:

Client IPv4: jimmy 192.10.10.3                                   port - 5000
          client index - 0  port - 0 flags - 0004

Client IPv4: portland 10.1.229.97                                port - 5000
          client index - 1  port - 32848 flags - 0005  connected
```

If the Sun SAM-Remote configuration includes several clients, press the CONTROL-f key sequence to scroll through each client. Each client is identified by name and by its client index field, an integer 0 - 9, which indicates its order in the possible 10 clients defined for this server daemon. The network block size, maximum file size, and minimum file size are listed in bytes. Flags indicate the state of the connection:

Flag	Number	Meaning
0x00000000	0004	No connection.
0xc0000000	0005	A connection has been established.

4. On the server, use the samu(1M) utility's v display to ensure that the Sun SAM-Remote catalog is available on the clients. From samu (1M), enter the following:

```
:v <eq>
```

For eq, specify the equipment ordinal of the Sun SAM-Remote client daemon as defined in the mcf file.

Example: **samu** (**v**) Display

The example shows the volumes that portland can access.

```
Robot VSN catalog by slot : eq 200 samu 4.0.5 Wed May 02 15:24:13
count 32
slot access time count use flags ty vsn
1 2003/01/02 10:40 0 0% -il-o-b-R-U- at 000032
2 2003/01/02 11:41 0 0% -il-o-b-R--- at 000034
3 2003/01/02 12:42 170 91% -il-o-b----- at 000035
4 2003/01/02 13:43 20 7% -il-o-b----- at 000037
5 2003/01/02 14:44 0 0% -il-o-b----- at 000038
6 2003/01/02 13:41 0 0% -il-o-b----- at 000031
```

5. From each client, issue the archiver(1M) command and its -A option to verify that archiving is taking place from the client to the server. This command writes a listing from the archiver, including the VSNs from the server. For information about this command, see the archiver(1M) man page.
If files are not archiving, see SAM-QFS Troubleshooting.

Recycling With the Sun SAM-Remote Software

This section contains information about recycling with Sun SAM-Remote. Sun Microsystems recommends recycling in a Sun SAM-Remote environment only under the very specific circumstances described here.

Because the recycling process involves freeing space on cartridges for more data, it is possible for the recycler to destroy important data on archive cartridges if the recycling process is not configured properly.

Note -
These restrictions are not enforced by the SAM-QFS software.

To avoid data loss, it is essential that you adhere to the following restrictions:

- Before using the recycler in a Sun SAM-Remote environment you must have a complete understanding of each step of the recycler. Executing commands in the wrong order, or on the wrong system, can result in an irreversible loss of data. Make sure you have analyzed a command's actions before executing any command, such as `tplabel`(1M), that can delete data on the Sun SAM-Remote client or the Sun SAM-Remote server.
- Recycling activities on the Sun SAM-Remote server and the Sun SAM-Remote client must not overlap. The result could be accidental relabeling of cartridges and irreversible loss of data.
- You must not recycle cartridges that contain removable media files.
- In a Sun SAM-Remote client and server environment, the client and server are unaware of each other's file systems, data files, and inode files. Therefore, the server and the client each must have exclusive use of a certain set of cartridges. Neither must ever use the other's cartridges.
  You can prevent accidental recycling of VSNs used by Sun SAM-Remote clients by creating a `no_recycle` list in the Sun SAM-Remote server's `/etc/opt/SUNWsamfs/ recycler.cmd` file. However, be careful of using the `chmed`(1M) command's `+c` option on volumes in a `no_recycle` list. When you use this command to set the recycling flag (`+c`) on a volume, that action overrides the `no_recycle` list in the `/etc/opt/SUNWsamfs/recycler.cmd` file.
- You must not attempt to recycle volumes on the Sun SAM-Remote server and Sun SAM-Remote client on the same day.

Recycling in a Sun SAM-Remote environment is allowed to occur only if the following conditions are present:

- Each VSN in the system is used by one client system or by the server. There cannot be files from multiple systems on any VSN.
- No Sun SAM-Remote client has catalog entries for any VSNs other than those VSNs containing that client's archive images. The regex values in the server configuration file's media definition lines (the eq media-type regex lines) must agree with the volumes specified in the client catalog. In addition, the regex specifications in the client catalogs cannot specify the same volumes.
- The archiving is performed on an archive set basis. When you are using Sun SAM-Remote, recycling must be performed by archive set, not by library.

The following subsections describe two methods for enabling recycling using a Sun SAM-Remote client and server. The methods are as follows:

- Recycling in a Sun SAM-Remote Environment--Method 1
- Recycling in a Sun SAM-Remote Environment--Method 2

Recycling in a Sun SAM-Remote Environment---Method 1

The procedures in this section describe one method for enabling recycling in a Sun SAM-Remote environment in which the server is named `sky` and the client is named `zeke`.

> Caution -
> To use the recycler in a Sun SAM-Remote environment, you must follow this procedure completely and you must test your configuration to verify that recycling is taking place correctly.

Configuration Files for the Server

The server must have Sun SAM-Remote configuration information in its `mcf` file and in its server configuration file. The following examples show these files.

Example: The **`mcf`** File on Server **`jimmy`**

```
# This is the mcf file for the server (jimmy).
# The server parameters file (rmt2000) points
# back to the correct automated library's equipment number
# (1000) for the SL8500 tape library.
#

# Equipment                                       Eq  Eq Family Dev Additional
# Identifier                                      Ord Ty Set    St  Parameters
# ==========                                      === == ====== == ==========
samfs1                                             100 ma samfs1 on
/dev/dsk/c6t600A0B80004850A600000F8048EF90ADd0s0  101 mm samfs1 on
/dev/dsk/c6t600A0B800048505600000E9D48EF91EEd0s6  102 mr samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE5F00048F2Dd0s6  103 mr samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE23000B24C2d0s6  104 mr samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE50000ADAECd0s6  104 mr samfs1 on

samfs2                                             200 ms samfs2 on
/dev/dsk/c7t60003BA13F71500048EDCE720001B17Fd0s6  201 md samfs2 on
/dev/dsk/c7t60003BA13F71500048EDCE7200014BEAd0s6  202 md samfs2 on

# SL8500
/etc/opt/SUNWsamfs/T10K        1000    sk      T10K    on      /etc/opt/SUNWsamfs/T10K_cat
/dev/rmt/4cbn                  1001    ti      T10K    on
/dev/rmt/5cbn                  1002    ti      T10K    on
/dev/rmt/0cbn                  1003    ti      T10K    on
/dev/rmt/1cbn                  1004    ti      T10K    on
/dev/rmt/6cbn                  1005    ti      T10K    on
/dev/rmt/7cbn                  1006    ti      T10K    on
/dev/rmt/2cbn                  1007    ti      T10K    on
/dev/rmt/11cbn                 1008    ti      T10K    on
/dev/rmt/10cbn                 1009    ti      T10K    on
/dev/rmt/12cbn                 1010    ti      T10K    on

# Define Sun SAM-Remote server jimmy
/etc/opt/SUNWsamfs/rmt2000 2000 ss jimmy on
```

Example: Server Configuration File on Server **jimmy**

```
# Server configuration file /etc/opt/SUNWsamfs/rmt2000 on jimmy.
# The eq of the automated library MUST match the eq of the
# automated library that you want to use in the mcf file.
tera
   media
     1000 ti 00002[0-9]
   endmedia
```

Configuration Files for Clients

The client must have Sun SAM-Remote configuration information in its `mcf` file and in its client configuration file. The following examples show these files.

Example: The **mcf** File on Client **tera**

```
# mcf file for client (tera)
#

# Equipment                                     Eq  Eq Family Dev Additional
# Identifier                                    Ord Ty Set    St  Parameters
# ==========                                    === == ====== == ==========
samfs1                                          100 ms samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE5F00048F2Dd0s6 101 md samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE23000B24C2d0s6 102 md samfs1 on
/dev/dsk/c7t60003BA13F71500048EDCE50000ADAECd0s6 103 md samfs1 on

# Define a L500 with 2 drives
/dev/samst/c4t500104F0009C2F6Fu0 300 rb L500 on
/dev/rmt/0cbn                    301 li L500 on
/dev/rmt/1cbn                    302 li L500 on


# Define tera as a Sun SAM-Remote client using jimmy as the server
/etc/opt/SUNWsamfs/rmt2000       2000 ss jimmy on
/dev/samrd/rd0                   2001 rd jimmy on
/dev/samrd/rd1                   2002 rd jimmy on
/dev/samrd/rd2                   2003 rd jimmy on
/dev/samrd/rd3                   2004 rd jimmy on
/dev/samrd/rd4                   2005 rd jimmy on
/dev/samrd/rd5                   2006 rd jimmy on
/dev/samrd/rd6                   2007 rd jimmy on
/dev/samrd/rd7                   2008 rd jimmy on
```

Example: Client Configuration File on Client **tera**

```
# cat /etc/opt/SUNWsamfs/rmt2000
# File /etc/opt/SUNWsamfs/rmt2000 on Sun SAM-Remote client tera: jimmy
```

How To Configure Recycling---Method 1

The procedure for configuring the recycling process includes a test for archiving and recycling. Because of the testing period, this procedure can take a day or two to complete, depending on how frequently files are archived and recycled.

> Note -
> Do not use the chmed(1M) command on the server to set the recycling flag (+c) for a client VSN. That action overrides the `no_recycle` list in the `/etc/opt/SUNWsamfs/recycler.cmd` file on the server.

Before starting the procedure, read About Recycling. Using the recycler in a Sun SAM-Remote environment requires a complete understanding of the steps in the recycling process. If you have not already familiarized yourself with the recycling process, do so now.

1. Make sure that the Sun SAM-Remote client and server are configured properly and that archiving is occurring.
   For more information, see Configuring the Sun SAM-Remote Software, which contains detailed information about configuring the Sun SAM-Remote client and server. That procedure includes steps for ensuring that archiving is taking place.
2. Edit the `archiver.cmd` file on the client system and add recycling directives.
   In this example, the recycling is performed by archive set, not by library. The directives specifying that recycling be done by archive set must appear in the `archiver.cmd` file.
   The example below shows the `archiver.cmd` file on client `zeke`. This file has been edited to communicate with the recycler.

   Example: **`archiver.cmd`** on a Client

```
# This is file /etc/opt/SUNWsamfs/archiver.cmd
# on Sun SAM-Remote client zeke.
#
archivemeta = off
archmax = li 12G

fs = samfs1
logfile = /var/adm/samfs1.archiver.log
no_archive tmp
all .
    1 -norelease 10m
    2 -norelease 10m

params
allsets -sort path -offline_copy stageahead -reserve set
allsets -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 2 -dataquantity 100G
allsets -ignore
allsets.1 -startage 8h -startsize 10G -drives 2
allsets.2 -startage 24h -startsize 20G -drives 2 -archmax 24G
endparams

vsns
all.1 li  ^10.*
all.2 li  ^20.*
endvsns
```

The directives shown in the example do the following:

- The `-recycle_hwm` directive sets the library's high-water mark for the archive set. When the utilization of the VSNs exceeds this percentage, recycling of the archive set begins.
- The `-recycle_ignore` directive is inserted only temporarily. This directive prevents recycling from occurring until you have configured and tested your environment. You can remove this directive in a later step.
- The `-recycle_mingain` directive is set high to ensure efficiency by limiting the amount of work needed to regain space.
- The `-recycle_vsncount 2` directive specifies that the recycler drain two VSN at a time. Do not let recycling overwhelm the system.

3. Edit the `recycler.cmd` file on the client and specify a log file to receive recycling log output.
   The example below shows the `recycler.cmd` file on client `zeke`, which has been edited to specify a recycler log file.

   Example: **`recycler.cmd`** on a Client

```
#
# This is the /etc/opt/SUNWsamfs/recycler.cmd file
# on client zeke.
#
logfile = /var/adm/recycler.log
```

4. Verify that the `archiver.cmd` file on the server is written to specify recycling by archive set.
   When using Sun SAM-Remote, you must specify that recycling be performed on an archive set basis, not by library. The directives specifying that recycling be done by archive set must appear in the `archiver.cmd` file.
   The following example shows the `archiver.cmd` file on server `sky`. This file specifies archiving by archive set.

   Example: **`archiver.cmd`** on a Server

```
# This is the archiver.cmd for the server (sky).
#
# Number of drives: 10
# Number of Mounted Filesystems: 1
# Number of Tests per Filesystem: 1
# Number of Archive Copies per Test: 2
#wait
#trace = /var/opt/SUNWsamfs/trace/archiver all
logfile = /var/opt/SUNWsamfs/log/archiver
interval = 1m
no_archive .
archmax = at 5G
drives = adic1 6
fs = samfs1
1 4h
testset testdir0
1 1m
2 1m
allsam1 .
1 1m
2 1m
params
allsam1.1 -drives 4 -drivemin 50m
allsam1.1 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
allsam1.1 -recycle_ignore
allsam1.2 -drives 4 -drivemin 50m
allsam1.2 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
allsam1.2 -recycle_ignore
testset.1 -drives 4 -drivemin 50m
testset.1 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
testset.1 -recycle_ignore
testset.2 -drives 4 -drivemin 50m
testset.2 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
testset.2 -recycle_ignore
endparams
vsns
samfs1.1 at 000000
allsam1.1 at 00000[1-5] # vsns 1 through 5.
allsam1.2 at 00000[6-9] # vsns 6 through 9.
testset.1 at 00001[0,4] # vsns 10 and 14.
testset.2 at 00001[5,9] # vsns 15 and 19.
endvsns
```

5. Edit the `recycler.cmd` file on the server. Modify the file to specify the following items:
   - A recycler log file to receive output from the recycler.
   - A `no_recycle` directive for the Sun SAM-Remote client's VSNs. The Sun SAM-Remote client is configured to write its copy 2 archive copies to cartridges in the Sun SAM-Remote server's library. The `no_recycle` directive is necessary to prevent the VSNs being used by the Sun SAM-Remote client for archiving from being recycled by the Sun SAM-Remote server.
     The following example shows the `recycler.cmd` file on server `sky`, which has been edited to specify a recycler log file.

   Example: **`recycler.cmd`** on a Server

   ```
   #
   # This is the /etc/opt/SUNWsamfs/recycler.cmd file
   # on Sun SAM-Remote server sky.
   #
   logfile = /var/opt/SUNWsamfs/recycler/recycler.log
   adic1 -ignore
   no_recycle at 00002[0-9] # Prevents VSNs assigned to zeke from
   # being recycled.
   ```

6. Use the `sam-recycler`(1M) command to test the recycler on the Sun SAM-Remote client.
   Run the recycler on the Sun SAM-Remote client system. This is a test to see if the recycler properly acknowledges the devices and VSNs specified in the configuration files.
   For example, you can use the following command to perform the initial test of the recycler:

```
zeke# sam-recycler -dvx
```

This testing is important, because if the recycler detects that the system on which it is running has no archive images on a particular VSN listed in any of that system's catalogs (including the historian catalog), the `recycler.sh` script can call for the cartridge to be labeled. Labeling a cartridge destroys all data on the cartridge. There is no communication between the Sun SAM-Remote client and the SAM-QFS servers to inform each side of the presence of archive copies. All such information is provided locally from local SAM-QFS file systems.
The recycler runs and logs its activity to the recycler log file. The recycler log file is defined in the `recycler.cmd` file. For more information about the `sam-recycler`(1M) command, see the `sam-recycler`(1M) man page.

7. Examine the recycler log file to find the following message:

```
Recycling is ignored on this archive set.
```

The following example shows a sample log file.

Example: Recycler Log File on a Client

```
# recycler.log from client zeke.
========== Recycler begins at Mon Jun 4 09:49:41 2001 ===========
Initial 7 catalogs:
0 Family: stk_l20 Path: /var/opt/SUNWsamfs/catalog/L20_cat
Vendor: STK Product: L20
SLOT ty capacity space vsn
0 lt 33.0G 33.0G 000173
1 lt 32.8G 44.1M CEL170
2 lt 33.0G 33.0G CEL139
4 lt 32.8G 16.8G CFC504
5 lt 33.0G 33.0G CFC503
6 lt 32.9G 0 CSM689
7 lt 32.9G 19.6G CSM690
8 lt 33.0G 33.0G CSM691
9 lt 33.0G 33.0G CSM692
10 lt 10.0G 10.0G CLN018
11 lt 33.0G 33.0G 000766
Total Capacity: 339.2G bytes, Total Space Available: 244.3G bytes
Volume utilization 27%, high 95% VSN_min 50%
Recycling is ignored on this robot.

1 Family: skyrs Path: /var/opt/SUNWsamfs/catalog/sky_cat
Vendor: (NULL) Product: (NULL)
SLOT ty capacity space vsn
0 at 48.5G 23.3G 000020
1 at 23.8G 23.8G 000021
2 at 48.5G 48.5G 000022
3 at 48.5G 48.5G 000023
4 at 48.5G 48.5G 000024
5 at 48.5G 2.6G 000025
6 at 48.5G 361.4k 000026
7 at 48.5G 48.5G 000027
8 at 48.5G 48.5G 000028
9 at 48.5G 0 000029
Total Capacity: 460.8G bytes, Total Space Available: 292.5G bytes
Volume utilization 36%, high 95% VSN_min 50%
Recycling is ignored on this robot.

2 Family: hy Path: /var/opt/SUNWsamfs/catalog/historian
Vendor: Sun SAM-FS Product: Historian
SLOT ty capacity space vsn
(no VSNs in this media changer)
Total Capacity: 0 bytes, Total Space Available: 0 bytes
Volume utilization 0%, high 95% VSN_min 50%
Recycling is ignored on this robot.

3 Family: defaultset.1 Path: /etc/opt/SUNWsamfs/archiver.cmd
```

```
Vendor: Sun SAM-FS Product: Archive set
SLOT ty capacity space vsn
0 lt 33.0G 33.0G 000766
1 lt 33.0G 33.0G 000173
2 lt 32.9G 0 CSM689
3 lt 32.9G 19.6G CSM690
4 lt 33.0G 33.0G CSM691
5 lt 33.0G 33.0G CSM692
Total Capacity: 197.6G bytes, Total Space Available: 151.5G bytes
Volume utilization 23%, high 60% VSN_min 90%
Recycling is ignored on this archive set.

4 Family: defaultset.2 Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS Product: Archive set
SLOT ty capacity space vsn
0 lt 32.9G 0 CSM689
1 at 48.5G 23.3G 000020
2 at 23.8G 23.8G 000021
3 at 48.5G 2.6G 000025
4 at 48.5G 361.4k 000026
5 at 48.5G 48.5G 000027
6 at 48.5G 48.5G 000028
7 at 48.5G 0 000029
Total Capacity: 348.0G bytes, Total Space Available: 146.8G bytes
Volume utilization 57%, high 60% VSN_min 90%
Recycling is ignored on this archive set.

5 Family: archiveset.1 Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS Product: Archive set
SLOT ty capacity space vsn
0 lt 32.8G 44.1M CEL170
1 lt 32.8G 16.8G CFC504
2 lt 33.0G 33.0G CFC503
Total Capacity: 98.6G bytes, Total Space Available: 49.8G bytes
Volume utilization 49%, high 60% VSN_min 90%
Recycling is ignored on this archive set.

6 Family: archiveset.2 Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS Product: Archive set
SLOT ty capacity space vsn
0 at 48.5G 23.3G 000020
1 at 23.8G 23.8G 000021
2 at 48.5G 48.5G 000022
3 at 48.5G 48.5G 000023
4 at 48.5G 48.5G 000024
Total Capacity: 218.0G bytes, Total Space Available: 192.8G bytes
Volume utilization 11%, high 60% VSN_min 90%
Recycling is ignored on this archive set.

21 VSNs:

---Archives--- -----Percent----- defaultset.1
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
in multiple sets 0 0 0 100 0 stk_l20:lt:CSM689
partially full 111 2.8G 8 31 61 stk_l20:lt:CSM690
empty VSN 0 0 0 0 100 stk_l20:lt:000173
empty VSN 0 0 0 0 100 stk_l20:lt:CSM691
empty VSN 0 0 0 0 100 stk_l20:lt:CSM692
empty VSN 0 0 0 0 100 stk_l20:lt:000766

---Archives--- -----Percent----- defaultset.2
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 skyrs:at:000029
no-data VSN 0 0 0 99 1 skyrs:at:000026
partially full 111 2.8G 6 88 6 skyrs:at:000025
empty VSN 0 0 0 0 100 skyrs:at:000028
empty VSN 0 0 0 0 100 skyrs:at:000027

---Archives--- -----Percent----- archiveset.1
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 99 1 stk_l20:lt:CEL170
partially full 677 2.3G 8 40 52 stk_l20:lt:CFC504
empty VSN 0 0 0 0 100 stk_l20:lt:CFC503

---Archives--- -----Percent----- archiveset.2
```

```
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
in multiple sets 0 0 0 51 49 skyrs:at:000020
empty VSN 0 0 0 0 100 skyrs:at:000022
empty VSN 0 0 0 0 100 skyrs:at:000023
empty VSN 0 0 0 0 100 skyrs:at:000024
in multiple sets 0 0 0 0 100 skyrs:at:000021

---Archives--- -----Percent----- stk_l20
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
empty VSN 0 0 0 0 100 stk_l20:lt:CLN018
partially full 13 80.3k 0 0 100 stk_l20:lt:CEL139
```

```
Recycler finished.
========== Recycler ends at Mon Jun 4 09:49:53 2001 ===========
```

8. Issue the `sam-recycler(1M)` command from the Sun SAM-Remote server to verify that the recycler is not recycling any VSNs reserved for the Sun SAM-Remote client.
   For example:

```
sky# sam-recycler -dvx
```

The preceding command runs the recycler and writes its activity to the recycler log file. For more information about the `sam-recycler(1M)` command, see the `sam-recycler(1M)` man page.
The following example shows a sample recycler log file.

Example: Recycler Log File on a Server

```
# recycler.log file from server sky.
========== Recycler begins at Mon Jun 4 09:50:44 2001 ===========
Initial 6 catalogs:
0 Family: adic1 Path: /var/opt/SUNWsamfs/catalog/adic1
Vendor: ADIC Product: Scalar 1000
SLOT ty capacity space vsn
0 at 1.3G 1.2G 000001
1 at 1.3G 1.3G 000002
2 at 1.3G 1.3G 000004
3 at 48.5G 0 000010
4 at 48.5G 0 000011
5 at 48.5G 43.5G 000018
6 at 48.5G 0 000019
7 at 48.5G 23.3G 000020
8 at 23.8G 23.8G 000021
9 at 48.5G 48.5G 000022
10 at 48.5G 48.5G 000023
11 at 48.5G 48.5G 000024
12 at 48.5G 2.6G 000025
13 at 48.5G 361.4k 000026
14 at 48.5G 48.5G 000027
15 at 48.5G 48.5G 000028
16 at 48.5G 0 000029
17 at 1.3G 1.3G 000005
18 at 48.5G 48.5G 000016
19 at 23.8G 23.8G CLN001
20 at 23.8G 23.8G CLN002
21 at 23.8G 23.8G CLN004
22 at 23.8G 23.8G CLN003
23 at 48.5G 421.6M 000015
24 at 1.3G 1.3G 000000
25 at 48.5G 0 000013
26 at 1.3G 1.3G 000003
27 at 48.5G 43.6G 000007
28 at 48.5G 41.8G 000008
29 at 48.5G 46.9G 000006
30 at 48.5G 48.3G 000009
31 at 48.5G 0 000014
32 at 48.5G 0 000012
33 at 48.5G 40.1G 000017
Total Capacity: 1.2T bytes, Total Space Available: 708.7G bytes
Volume utilization 43%, high 95% VSN_min 50%
Recycling is ignored on this robot.

1 Family: hy Path: /var/opt/SUNWsamfs/catalog/historian
Vendor: Sun SAM-FS Product: Historian
SLOT ty capacity space vsn
(no VSNs in this media changer)
Total Capacity: 0 bytes, Total Space Available: 0 bytes
Volume utilization 0%, high 95% VSN_min 50%
Recycling is ignored on this robot.
```

```
2 Family: testset.1 Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS Product: Archive set
SLOT ty capacity space vsn
0 at 48.5G 0 000010
1 at 48.5G 0 000014
Total Capacity: 97.1G bytes, Total Space Available: 0 bytes
Volume utilization 100%, high 60% VSN_min 90%: *** Needs recycling ***
Recycling is ignored on this archive set.

3 Family: testset.2 Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS Product: Archive set
SLOT ty capacity space vsn
0 at 48.5G 0 000019
1 at 48.5G 421.6M 000015
Total Capacity: 97.1G bytes, Total Space Available: 421.6M bytes
Volume utilization 99%, high 60% VSN_min 90%: *** Needs recycling ***
Recycling is ignored on this archive set.

4 Family: allsam1.1 Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS Product: Archive set
SLOT ty capacity space vsn
0 at 1.3G 1.2G 000001
1 at 1.3G 1.3G 000002
2 at 1.3G 1.3G 000004
3 at 1.3G 1.3G 000005
4 at 1.3G 1.3G 000003
Total Capacity: 6.5G bytes, Total Space Available: 6.3G bytes
Volume utilization 3%, high 60% VSN_min 90%
Recycling is ignored on this archive set.

5 Family: allsam1.2 Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS Product: Archive set
SLOT ty capacity space vsn
0 at 48.5G 43.6G 000007
1 at 48.5G 41.8G 000008
2 at 48.5G 46.9G 000006
3 at 48.5G 48.3G 000009
Total Capacity: 194.2G bytes, Total Space Available: 180.6G bytes
Volume utilization 6%, high 60% VSN_min 90%
Recycling is ignored on this archive set.

Need to select candidate for media changer testset.1 to free up 39.8G bytes.
Quantity of data to move limited to (no limit) bytes and 1 VSNs.
Checking 000010. Need to free 39.8G, quantity limit: (no limit), VSN count: 1.
VSN is in correct media changer... good.
VSN is not already recycling... good.
VSN has no request files... good.
VSN has no 'archive -n' files...good.
VSN was not specified as "no_recycle" in recycler.cmd file... good.
VSN does not exceed VSN count limit... good.
VSN does not exceed data quantity limit... good.
VSN meets minimum gain requirement.
Recycling is ignored on this media changer - VSN not marked for recycling.

Checking 000014. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN is in correct media changer... good.
VSN is not already recycling... good.
VSN has no request files... good.
VSN has no 'archive -n' files...good.
VSN was not specified as "no_recycle" in recycler.cmd file... good.
VSN exceeds VSN count limit - skipped.
Checking 000019. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000015. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking 000001. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000003. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000004. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000005. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
```

```
Checking 000002. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000008. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000007. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking 000006. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000009. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000011. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000029. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000013. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000012. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking 000026. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000025. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000020. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000017. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000018. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN003. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking 000021. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000022. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000027. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000028. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000023. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000024. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000016. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking CLN001. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN002. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN004. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000000. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
No candidate was found in this media changer.

Need to select candidate for media changer testset.2 to free up 38.8G bytes.
Quantity of data to move limited to (no limit) bytes and 1 VSNs.
Checking 000010. Need to free 38.8G, quantity limit: (no limit), VSN count: 1.
VSN not in correct media changer.
Checking 000014. Need to free 38.8G, quantity limit: (no limit), VSN count: 1.
VSN not in correct media changer.

Checking 000019. Need to free 38.8G, quantity limit: (no limit), VSN count: 1.
VSN is in correct media changer... good.
VSN is not already recycling... good.
VSN has no request files... good.
VSN has no 'archive -n' files...good.
VSN was not specified as "no_recycle" in recycler.cmd file... good.
VSN does not exceed VSN count limit... good.
VSN does not exceed data quantity limit... good.
VSN meets minimum gain requirement.
Recycling is ignored on this media changer - VSN not marked for recycling.
```

```
Checking 000015. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN is in correct media changer... good.
VSN is not already recycling... good.
VSN has no request files... good.
VSN has no 'archive -n' files...good.
VSN was not specified as "no_recycle" in recycler.cmd file... good.
VSN exceeds VSN count limit - skipped.

Checking 000001. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000003. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000004. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000005. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000002. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking 000008. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000007. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000006. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000009. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000011. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000029. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking 000013. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000012. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000026. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000025. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000020. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000017. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking 000018. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN003. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000021. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000022. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000027. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000028. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000023. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.

Checking 000024. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000016. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN001. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN002. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN004. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000000. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
No candidate was found in this media changer.
34 VSNs:
```

```
---Archives--- -----Percent----- testset.1
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 adic1:at:000010
no-data VSN 0 0 0 100 0 adic1:at:000014

---Archives--- -----Percent----- testset.2
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 adic1:at:000019
partially full 677 2.3G 5 93 2 adic1:at:000015

---Archives--- -----Percent----- allsam1.1
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
partially full 97 173.8M 1 9 90 adic1:at:000001
no-data VSN 0 0 0 2 98 adic1:at:000003
no-data VSN 0 0 0 2 98 adic1:at:000004
empty VSN 0 0 0 0 100 adic1:at:000005
empty VSN 0 0 0 0 100 adic1:at:000002

---Archives--- -----Percent----- allsam1.2
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 13 87 adic1:at:000008
partially full 98 1.6G 3 7 90 adic1:at:000007
no-data VSN 0 0 0 3 97 adic1:at:000006
empty VSN 0 0 0 0 100 adic1:at:000009

---Archives--- -----Percent----- adic1
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 adic1:at:000011
no_recycle VSN 0 0 0 100 0 adic1:at:000029
no-data VSN 0 0 0 100 0 adic1:at:000013
no-data VSN 0 0 0 100 0 adic1:at:000012
no_recycle VSN 0 0 0 99 1 adic1:at:000026
no_recycle VSN 0 0 0 94 6 adic1:at:000025
no_recycle VSN 0 0 0 51 49 adic1:at:000020
no-data VSN 0 0 0 17 83 adic1:at:000017
no-data VSN 0 0 0 10 90 adic1:at:000018
empty VSN 0 0 0 0 100 adic1:at:CLN003

no_recycle VSN 0 0 0 0 100 adic1:at:000021
no_recycle VSN 0 0 0 0 100 adic1:at:000022
no_recycle VSN 0 0 0 0 100 adic1:at:000027
no_recycle VSN 0 0 0 0 100 adic1:at:000028
no_recycle VSN 0 0 0 0 100 adic1:at:000023
no_recycle VSN 0 0 0 0 100 adic1:at:000024
empty VSN 0 0 0 0 100 adic1:at:000016
empty VSN 0 0 0 0 100 adic1:at:CLN001
empty VSN 0 0 0 0 100 adic1:at:CLN002
empty VSN 0 0 0 0 100 adic1:at:CLN004
partially full 12 88.3k 0 0 100 adic1:at:000000
```

```
Recycler finished.
========== Recycler ends at Mon Jun 4 09:51:05 2001 ===========
```

9. Analyze the server and client `recycler.log` files to choose VSNs that are candidates for recycling.
   Near the end of the `recycler.log` file is a `Status` column.
   - In the client log files, VSNs with the following types of status entries are candidates for recycling:
     - `no-data VSN`. To recycle a `no-data` VSN, see To Recycle `no-data` VSNs.
     - `partially full`. To recycle a `partially full` VSN, see To Recycle `partially full` VSNs.
   - In the server log file, the best candidates for recycling are those with a 0 value in the `Count`, `Bytes`, and `Use` columns.

How To Recycle **`no-data`** VSNs

The `no-data` VSNs are the easiest VSNs to recycle. For these, the `Count`, `Bytes`, and `Use` field values are all 0 (zero).

1. Examine the `recycler.log` file from the client to see if there are any `no-data` VSNs.
   VSNs 000029 and 000026 from the client `zeke` can be considered for recycling because they are `no-data` VSNs, as shown in the code below.

```
# From the client zeke recycler.log file:
---Archives--- -----Percent----- defaultset.2
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 skyrs:at:000029
no-data VSN 0 0 0 99 1 skyrs:at:000026
partially full 111 2.8G 6 88 6 skyrs:at:000025
empty VSN 0 0 0 0 100 skyrs:at:000028
empty VSN 0 0 0 0 100 skyrs:at:000027
```

2. Examine the `recycler.log` file from the server and determine if the VSNs you selected from the previous step are represented identically in the server's recycler log file.
   Verify that there is no active data from the server archived on those VSNs.
   The code below shows the data for the `no_recycle` VSNs in the server's `recycler.log` file. For VSNs 000029 and 000026, the data in the server's `recycler.log` file is identical to that in the client's `recycler.log` file.

```
# From the Server log file:
---Archives--- -----Percent----- adic1
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 adic1:at:000011
no_recycle VSN 0 0 0 100 0 adic1:at:000029
no-data VSN 0 0 0 100 0 adic1:at:000013
no-data VSN 0 0 0 100 0 adic1:at:000012
no_recycle VSN 0 0 0 99 1 adic1:at:000026
no_recycle VSN 0 0 0 94 6 adic1:at:000025
no_recycle VSN 0 0 0 51 49 adic1:at:000020
no-data VSN 0 0 0 17 83 adic1:at:000017
no-data VSN 0 0 0 10 90 adic1:at:000018
empty VSN 0 0 0 0 100 adic1:at:CLN003
.
.
.
```

3. If no active data from the server is archived on the selected VSNs, use the `tplabel`(1M) or `odlabel`(1M) command to relabel the VSNs.

   > Note -
   > This destroys all data on the VSN and reclaims space.

   For example, for tape VSN 000029, use the following command:

```
sky# tplabel -vsn 000029 -old 000029 at.000029
```

When this VSN 000029 is relabeled, you regain 100 percent of the space on that VSN.
If the media is a magneto-optical disk, use the odlabel(1M) command. For more information, see the odlabel(1M) man page.

How To Recycle **partially full** VSNs

The VSNs for which a partially full status is reported can also be recycled.

1. Examine the recycler.log file from the client to see if there are any partially full VSNs.
   You can consider VSN 000025 from the client, zeke, for recycling because its status is partially full, as shown below.

```
# From the client zeke recycler.log file:
---Archives--- -----Percent----- defaultset.2
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 skyrs:at:000029
no-data VSN 0 0 0 99 1 skyrs:at:000026
partially full 111 2.8G 6 88 6 skyrs:at:000025
empty VSN 0 0 0 0 100 skyrs:at:000028
empty VSN 0 0 0 0 100 skyrs:at:000027
```

   VSN 000025 shows that 6 percent of its space is in use. These are active archive images that must be rearchived before this VSN can be recycled. The following steps in this process show how to ensure that these active archive images are rearchived to another VSN.

2. Examine the recycler.log file from the server side to ensure that no active data from the server is archived on that VSN.
   The server's recycler.log file indicates that VSN 000025 is 6 percent free, which is the same percentage that was reported in the client's recycler.log file. The server is not aware of the client's archive images, so it reports that all of the remaining 94 percent is consumed by obsolete archive images.

```
# From the Server log file:
---Archives--- -----Percent----- adic1
-----Status----- Count Bytes Use Obsolete Free Library:Type:VSN
no-data VSN 0 0 0 100 0 adic1:at:000011
no_recycle VSN 0 0 0 100 0 adic1:at:000029
no-data VSN 0 0 0 100 0 adic1:at:000013
no-data VSN 0 0 0 100 0 adic1:at:000012
no_recycle VSN 0 0 0 99 1 adic1:at:000026
no_recycle VSN 0 0 0 94 6 adic1:at:000025
no_recycle VSN 0 0 0 51 49 adic1:at:000020
no-data VSN 0 0 0 17 83 adic1:at:000017
.
.
.
```

3. On the server, use the chmed(1M) command with the +c option to rearchive the active files on the VSN.

```
sky# chmed +c at.000025
```

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

4. On the client, use the sam-recycler(1M) command to run the recycler again.

```
zeke# sam-recycler -dvx
```

   This marks each active file to be rearchived to another VSN.

5. Start the archiver.
   You can do this either by letting the archiver run normally, or by typing :arrun from the samu(1M) utility on the client. For more information about the :arrun command, see the samu(1M) man page.
6. When archiving is complete, issue the sam-recycler(1M) command to rerun the recycler on the client.

```
zeke# sam-recycler -dvx
```

This ensures that all active files have been rearchived.

7. If the Count, Bytes, and Use field values are all 0 (zero), use the `tplabel`(1M) or `odlabel`(1M) command to relabel the VSN from the server.
   For example, for tape VSN 000025, use the following command:

```
sky# tplabel -vsn 000025 -old 000025 at.000025
```

   This command relabels the VSN and destroys all data on it. After this VSN is relabeled, you regain 88 percent of the space on this VSN.
   If the media had been a magneto-optical disk, you would have used the `odlabel`(1M) command. For more information about the `odlabel`(1M) command, see the `odlabel`(1M) man page.

Recycling in a Sun SAM-Remote Environment---Method 2

This section presents another way you can recycle volumes using Sun SAM-remote software.

> Caution -
> Use the recycler in a Sun SAM-Remote environment only after following the steps in this procedure completely and only after testing your configuration to verify that recycling is taking place correctly.

How To Configure Recycling---Method 2

1. On the Sun SAM-Remote client, issue the `sam-recycler`(1M) command to determine which volumes are the best candidates for recycling. For example:

```
client# sam-recycler -dvx
```

2. Analyze the recycler log file for recycling candidates.
   Toward the end of the `recycler.log` file is a `Status` column. VSNs with the following types of status entries in the client log file are candidates for recycling:
   - `no-data VSN`. To recycle a `no-data` VSN, see To Recycle `no-data` VSNs.
   - `partially full`. To recycle a `partially full` VSN, see To Recycle `partially full` VSNs.
     In the server log file, the best candidates for recycling are those with a 0 value in the `Count`, `Bytes`, and `Use` columns.
3. On the Sun SAM-Remote server, issue the `chmed`(1M) command to set the recycle flag on the selected VSNs.
   For example:

```
server# chmed +c at.00025
```

4. Wait until the VSNs being recycled are drained completely of archive images.
   The archiver on the client side does this.
5. On the Sun SAM-Remote server, issue the `tplabel`(1M) or `odlabel`(1M) command, depending on the archive media, to relabel the volumes.
6. On the Sun SAM-Remote server, clear any flags, such as `R` or `c` that prevent the volumes from being used for archiving on the Sun SAM-Remote client.

Using SAM-QFS on Linux Clients

Contents

Using SAM-QFS on Linux Clients

This document contains important information about Linux clients in the Sun QFS file system in Release 5.0. The document is written for system administrators and programmers who are already familiar with the Sun QFS software running on the Solaris™ Operating System (OS). It explains the differences between installation and configuration on the Solaris OS and installation and configuration on the Linux OS.

Before installing the software on a Linux client, you should be familiar with the detailed installation procedure for software as outlined in Installing Sun QFS. You should also have an experienced system administrator's knowledge of the supported Red Hat Enterprise or SuSE Linux operating environment in which you are installing the software.

About Shared File Systems and the Linux Client

A shared file system is a distributed, multihost file system that you can mount on multiple hosts. One Solaris OS host acts as the metadata server, and the others act as clients. You can also designate one or more Solaris clients as potential metadata servers, enabling you to switch metadata servers.

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

The file system functionality is largely the same for the Solaris and Linux clients. Software components are stored in the `/opt/SUNWsamfs`, `/etc/opt/SUNWsamfs`, and `/var/opt/SUNWsamfs` directories on the Linux client, just as they are on a Solaris system. The `sam-fsd` and `sam-sharefsd` commands function the same way on both platforms.

Within a shared file system, the Sun QFS software can be installed on Linux clients as well as on Solaris clients. The Sun QFS Linux client software supports the following Linux distributions on x64 platforms:

- Red Hat Enterprise Linux 4.0 (UD-4)
- Red Hat Enterprise Linux 4.5
- SuSE Linux Enterprise Server 9 (service pack 2)
- SuSE Linux Enterprise Server 10
- SuSE Linux Enterprise Server 10 (service pack 2)

Installing and Uninstalling the Linux Client

This section provides instructions for installing and uninstalling the Sun QFS Linux Client software.

The following are the differences in installation between the Solaris client and the Linux client:

- The Linux software comes in two folders on the DVD or in the download packages. The installation files can be found in the `linux1` directory on that DVD (or in the downloaded software package).
- The Linux installation packages are in the form of RPMs (Red Hat Package Manager) and SRPMs (Source RPM). RPMs typically contain binary commands and loadable modules, whereas SRPMs contain source code RPMs.
- Because so many variations of the Linux kernel exist, the installation program also includes custom kernel capability, which enables the software to work with as many kernel variations as possible.

How to Install the Linux Client

1. Edit the Solaris metadata server's `/etc/opt/SUNWsamfs/hosts.}}fsname file to add the Linux client's address, and run {{samsharefs` to update the information on disk.
   In the following example, `hiball-mn` is the Solaris metadata server:

```
hiball-mn 10.1.170.213 1 0 server
linux-mn 10.1.170.210 0 0
rollerball-mn 10.1.170.132 0 0
sandiego-mn 10.1.170.8 0 0
```

2. Insert the installation DVD into the Linux system or locate the downloaded software package.
3. Do one of the following to install the software:
   - If you installing from the DVD, type the following commands as `root` on the Linux system:

     ```
     # mount -o ro -t iso9660 /dev/cdrom /mnt
     # /mnt/linux1/Install
     ```

   - If you are installing from a downloaded software package, type the following commands as `root` on the Linux system:

     ```
     # mount -o ro,loop -t iso9660 StorageTek_QFS_5.0.iso /mnt
     # /mnt/linux1/Install
     ```

     The installation program installs the software.
     If the installation program does not recognize the Linux kernel version, it will display the following message:

     ```
     A direct match for your kernel wasn't found. Attempt creating a custom rpm for your
     kernel (yes/no)?
     ```

     Select `yes` to enable the installation program to adapt the loadable binary kernel module to the Linux system. If you select `no`, the installation cannot be completed.
4. Review the `/etc/opt/SUNWsamfs/mcf` file to verify that it contains the correct paths for your installation.

   > Note -
   > An `mcf` file is automatically generated for the Linux client. For more information, see `mcf` File Differences.

   For most installations, the `mcf` file that is created will be correct, but it may have to be edited to reflect your specific environment.
5. Verify that the Solaris metadata server is running, and then run the `mount` command to mount the file system.

   > Note -
   > The `mount` command requires that the `shared` mount option be set, either on the command line or in `/etc/fstab`.

   When you mount the file system, the following message is displayed:

   ```
   Warning: loading SUNWqfs will taint the kernel: SMI license
   See http://www.tux.org/lkml/#export-tainted for information
   about tainted modules. Module SUNWqfs loaded with warnings
   ```

   This is a normal result of the installation and can be disregarded.

How to Configure Access to **man** Pages

Sun QFS man pages are located in section 1m. In order to ensure access to the man pages from the Linux client, carry out the following procedures:

- On Red Hat Linux clients, add `1m` to `MANSECT` in the `/etc/man.config` file.

- On SuSE clients, add `1m` to `SECTION` in the `/etc/manpath.config` file.

How to Uninstall the Linux Client Software

1. To uninstall the Linux client software, use the uninstall script that resides in the `/var/opt/SUNWsamfs` directory.

> Caution -
> Do not use other processes, such as `rpm -e`, to uninstall the software. They can cause unexpected results and problems with uninstalling or reinstalling the software.

Linux Differences

The following sections describe the functionality, mounting, and execution differences between the Linux and Solaris clients.

Differences in Function

There are a few areas in which the Linux client does not match the functionality of the Solaris client. The Linux client does not support the following functions:

- Remote file sharing
- `samaio`
- Quotas
- Access control lists (ACLs)
- Buffer-cache readahead
- Browser interface (BI) support
- Forced unmount
- IPv6

The Linux client supports striping, but striped performance is generally limited to the performance of a single device.

`mcf` File Differences

The `mcf` file, `/etc/opt/SUNWsamfs/mcf`, defines the topology of the equipment that the file system manages.

The Linux client software automatically generates an `mcf` file. If an `mcf` file does not exist, the Linux client will create one when the system is booted or when `samd config` is run. This file should be examined to verify that it correctly reflects the Sun QFS environment. It may need to be manually edited to match your specific file system configuration. Note, however, that if you create your own `mcf` file or edit the auto-generated mcf file, the system will no longer auto-generate the file.

When an existing disk is repartitioned or a new file system is created, run `samd config` or reboot the system to create a new `mcf` file that reflects the changes. When new hardware is added, the SCSI bus needs to be rescanned with the “add single device” mechanism, or the equivalent, and then `samd config` must be run. Rebooting the system will also cause the SCSI bus to be rescanned.

If you need to modify the Linux client `mcf` file, run the `samfsconfig` command first. This prints the existing file system names and the Linux device path names that are associated with them. When editing the `mcf` file, you must take into account the differences between Solaris and Linux logical unit numbers (LUNs) in the device paths.

Instead of specifying an actual device path for the metadata device (mm), specify `nodev` in the Linux `mcf` file. This is the default in the auto-generated `mcf` file.

```
belmont 40 ma belmont on shared
nodev 43 mm belmont on
/dev/sdb5 44 mr belmont on
```

For more information about creating `mcf` files, see the `mcf`(4) man page or About the Master Configuration File.

Mounting Differences

Mounting functions are almost identical on the Linux and Solaris clients. However, the mount options available for the Linux client are limited to the following:

- `rw`, `ro`
- `retry`
- `shared`
- `meta_timeo`
- `rdlease`, `wrlease`, `aplease`
- `minallocsz`, `maxallocsz`
- `min_pool` (ignored outside of `samfs.cmd`)
- `noauto`, `auto` (ignored outside of `/etc/fstab`)

No other mount options are available for the Linux client.

Execution Differences

The Linux application programming interface (API) differs from the Solaris API in many respects, so users will see some differences. For example, upon lookup of a directory that has been removed, the Solaris system returns an ENOENT message, whereas the Linux system returns an ESTALE message.

There are several host bus adapter (HBA) failover methods that are supported in the various Linux distributions, and failover is included in QLogic drivers. None of these methods have been exhaustively tested with the Linux client.

> Caution -
> Do not use the `mdadm` (multiple devices admin) package for path failover on a Linux client. The `mdadm` package writes a superblock to devices that it uses. The result is that `mdadm` has the potential to corrupt data that Solaris has written to those devices. Furthermore, Solaris can also corrupt the superblock that `mdadm` has written to the devices.

Performance Differences

Performance of the Linux client is affected by the fact that direct I/O is more restrictive on the Linux client. The Linux device layer supports a maximum of 4 kilobytes per I/O operation. This means that large requests in a Linux environment must be broken up and serviced individually.

Linux Kernel Patch

Solaris 10 added a new partition ID for storage management initiative (SMI) labels on x86 and x64 platforms. Linux kernels with versions before 2.6.10 do not recognize this partition ID, so those kernels will need a patch in order to support Sun QFS Linux client interaction with a Sun QFS metadata server running Solaris 10 on an x64 platform.

A patch is included with the Sun QFS Linux client software. It is located in the `linux1/patches` directory. To install the patch, navigate to the top level of the Linux kernel source directory and run the `patch` command. For example:

```
# cd /usr/src/linux
# patch -p1 < /<path-to-patch>/linux_2.6_smi.patch
```

SELinux

Unexpected results may occur when you run the Sun QFS Linux client software on a system with SELinux enabled, depending on your site's specific SELinux configuration. For example, the `df` command may return `Permission denied` when run by a non-root user.

If you encounter such a similar error, check the security context settings on the `/dev/samsys` file. The correct settings should be similar to the following:

`crw-r-r-` 1 user_u:object_r:device_t root root 253, 0 Jan 31 17:19 /dev/samsys

You might also want to check the security label of the user and verify that the user has a sufficient security label to access the object, given the object's security label.

Troubleshooting

Linux clients and Solaris clients use different procedures to store system information that might be used to diagnose Sun QFS issues.

Files that contain system information from the Linux kernel are in the `/proc` directory. For example, the `/proc/cpuinfo` file contains hardware information. The table below describes some files that contain useful troubleshooting information.

Table – Linux Files for Troubleshooting

File Name	Information Provided
`version`	Running kernel version
`cpuinfo`	Hardware information
`uptime`	Time in seconds since boot time, and total time used by processes
`modules`	Information about the modules that are loaded
`cmdline`	Command-line parameters that are passed to the kernel at boot time
`filesystems`	Existing file system implementations
`scsi/scsi`	Attached SCSI devices
`fs/samfs/<QFS file system>/fsid`	File system ID, which must be included in the share options for network file system (NFS)

> Note -
> Linux kernel log messages go to the `/var/log/messages` file.

Troubleshooting Tools

Because the Linux kernel has many variations, troubleshooting problems can be very challenging. A few tools are available that might help in debugging:

- Projects such as `lkcd` and `kgdb` can provide kernel crash dump information.
- Projects such as `kdb`, `kgdb`, and `icebox` are kernel debuggers.

> Note -
> These projects are not present by default in Red Hat or SuSE Linux. You must obtain the appropriate RPMs or SRPMs and might have to reconfigure the kernel to use them.

- The `strace` command traces system calls and signals. It is similar to the Solaris `truss` command.
- The Sun QFS `samtrace` command dumps the trace buffer.
- The Sun QFS `samexplorer` command generates a Sun QFS diagnostic report script.

> Note -
> Trace files are placed in the `/var/opt/SUNWsamfs/trace` directory on the Linux client, just as they are on the Solaris client.

Frequently Asked Questions

The following are some questions about the Linux client that are frequently asked by users who are familiar with Sun QFS file systems on the Solaris platform.

Q: The Linux install script reports that I got a negative score and cannot install the software. Is there any way I can still install the software?

A: You can try the `-force-custom` and `-force-build` install options. However, this may cause a system panic when installing the modules.

This is especially risky if your kernel is built with some of the kernel hacking options, such as spinlock debugging, enabled.

Q: Can I use commands such as `vmstat`, `iostat`, `top`, `truss`, and `sar` on Linux?

A: The `vmstat`, `top`, `sar`, and `iostat` commands are found in many Linux Red Hat installations. If they are not installed, they can be added using the `sysstat` and `procps` RPMs. The Linux equivalents of `truss` are `ltrace` and `strace`.

Q: How can multipath failover be done with QFS on Linux?

A: If you are using the Sun StorageTek 6130, you can use the multipath tool found at http://www.sun.com/download/products.xml?id=432f43a5. Otherwise, several path failover methods are possible, depending on your Linux distribution and type of HBAs. These include but are not limited to Sun StorageTek Traffic Manager (SSTM), multipulse, device mapper, and qlogic path failover. None of these have been tested exhaustively with Sun QFS.

Do not use the `mdadm` (multiple devices admin) package for path failover on a Sun QFS Linux client. The `mdadm` package writes a superblock to devices that it uses. The result is that `mdadm` has the potential to corrupt data that Solaris has written to those devices. Furthermore, Solaris can also corrupt the superblock that `mdadm` has written to the devices.

Q: Can Sun StorageTek™ Traffic Manager be used with the Sun QFS Linux client?

A: Yes. First build a custom kernel with multipathing support as described in the Sun StorageTek Traffic Manager documentation. Then install the Linux client software.

Q: Can Extensible Firmware Interface (EFI) labels be used on the Sun QFS Linux client?

A: Most Linux Red Hat kernels are not built with support for EFI labels with GPT (GUID Partition Table) partitions. Therefore, to use EFI labels, you must rebuild the kernel with the CONFIG_EFI_PARTITION option set. For more information about building a custom kernel, see the distribution documentation.

Q: Can I use other Linux volume managers such as logical volume management (LVM), Enterprise Volume Management System (EVMS), or Device Mapper with the Sun QFS Linux client software?

A: No. To use a file system with EVMS, you need to have a File System Interface Module (FSIM) for that file system. No FSIM exists for the Sun QFS product. For you to use LVM, the partition type that `fdisk` shows must be LVM(8e). Partitions that Sun QFS uses must be SunOS.

Q: Can I use file systems that are larger than 2 TB?

A: Yes, but some utilities that provide file system information, such as `df`, might return incorrect information when run on Linux. In addition, there may be problems when sharing the file system with NFS or Samba.

Q: Are there any differences between the mount options supported on the Linux client and those supported on the Solaris client?

A: There are many `samfs` mount options that are not supported on the Linux client. Two to be aware of are `nosuid` and `forcedirectio`. See Mounting Differences for a complete list of supported mount options on the Linux client.

Q: How can I mount filesystems that have SMI labels generated by X64 Solaris 10 metadata server?

A: Apply one of the patches available from Sun for the 2.4 or 2.6 kernel.

Q: What Linux versions and distributions were used in testing Sun QFS 5.0?

A: Sun QFS was tested with the following Linux distributions:

- 2.4.21-32 on x86 and X64 - RH 3 Update 5
- 2.4.21-47 on x86 and X64 - RH 3 Update 8
- 2.6.16-8 on X64 - RH 4 Update 2
- SuSE 8 SP4
- 2.6.5-7.191 on X64 and Altix - SuSE 9 SP2
- 2.6.16.21-0.8 - SuSE 10 FCS

Q: The SAM-QFS Manager user interface does not show Sun QFS installed on my Linux clients. Why?

A: The SAM-QFS Manager software does not support operations on Linux clients.

Q: Is there a Linux version of `cfgadm` or `devfsadm` to rescan/reprobe the SCSI devices?

A: On Linux you must reboot the system in order to guarantee that the SCSI devices are rescanned. You can also try removing and reinstalling the HBA driver. For example:

```
rmmod qlaXXXX; modprobe qlaXXXX
```

If you are using a Qlogic driver that supports LUN hot add, you can try the following:

`echo "scsi-qlascan" > /proc/scsi//` (qlogic driver will re-scan)

Q: There's no forced unmount on the Linux client like there is on Solaris. What can I do when I encounter a busy filesystem when attempting to unmount?

A: First try `fuser -k`. If the busy filesystem message persists, use `lsof` to find any open files and their associated PIDS, then kill the processes.

Using SAM-QFS With Solaris Cluster

Contents

Using SAM-QFS With Solaris Cluster

You can use Sun QFS 5.1 file systems with Solaris Cluster 3.2 1/09 software to provide failover and high availability solutions. When used with Solaris Cluster, Sun QFS file systems are limited to fewer nodes than without Solaris Cluster.

- On SPARC systems, Sun QFS with Solaris Cluster supports up to 16 nodes.
- On x64 systems, Sun QFS with Solaris Cluster supports up to 8 nodes.

> Note -
>
> - Online grow and shrink are not supported with Solaris Cluster software.
> - Solaris Cluster was previously known as Sun Cluster.
> - You can view all the information about working with Solaris Cluster on a single long page.

About Failover and Highly Available Configurations

You can install a Sun QFS file system in a Solaris Cluster environment and configure the file system for failover or high availability. The following configuration methods are available, depending on whether your file system is shared or unshared:

- Unshared file system - A standalone, local file system configured in a Solaris Cluster environment is considered to be a highly available local file system. When the node hosting the file system fails, the Solaris Cluster software moves the file system to another node. The highly available QFS file system uses the `SUNW.HAStoragePlus` resource type, which is distributed with the Solaris Cluster product. From the Solaris Cluster perspective, this use is referred to as a failover file system.

- Shared file system - In this configuration, one of the Solaris Cluster nodes acts as the metadata server for the file system. If the metadata server fails, the Solaris Cluster software automatically moves file system operations from the failing metadata server to a different metadata server within the cluster. Because the file system data is already shared, no additional movement is required. The shared file system configuration uses the `SUNW.qfs` resource type, which is distributed with the Sun QFS product. From the Solaris Cluster perspective, this use is simply known as a clustered file system.

> Note -
>
> - The Oracle Real Application Clusters (RAC) application is the only scalable application that the Sun QFS software supports in Solaris Cluster environments. To configure an Oracle database with Sun QFS and Solaris Cluster software, see the Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS.
> - You can also have clients that are part of the shared file system but that are not part of the Solaris Cluster configuration. For more information, see Configuring Clients Outside of the Cluster.

- Shared archiving - You can also configure SAM-QFS archiving features for high availability with Solaris Cluster software (HA-SAM).

When an unrecoverable problem occurs in an HA-SAM configuration, Solaris Cluster software moves the archiving and staging operations to a healthy node. HA-SAM depends on the Sun QFS Solaris Cluster agent, so this configuration must be installed with a shared file system that is mounted and managed by the Sun QFS Solaris Cluster agent. For more information, see Configuring SAM-QFS in a Sun Cluster Environment (HA-SAM).

> Note -
> Although installing the file system in a Solaris Cluster environment improves reliability and decreases or eliminates unplanned downtime, it does not eliminate planned downtime. To maintain the health of the file system, you might still need to bring down the Sun QFS or SAM-QFS software occasionally to run the `samfsck` process. You will also need to shut down the software to apply software patches or updates.

Task Map: Working With Solaris Cluster Software

Enabling Solaris Cluster software to work with Sun QFS or SAM-QFS involves the following tasks:

Step	Task	Description
1	Confirm Solaris Cluster-specific requirements.	The basic requirements for Sun QFS or SAM-QFS with Solaris Cluster software are very similar to those for the products without Solaris Cluster software. However, there are some specific requirements that will simplify later steps in the process. For information, see Requirements for Using SAM-QFS With Solaris Cluster.
2	Install Solaris Cluster software and configure nodes as appropriate.	If not already done, install and configure Solaris Cluster software as explained in the Solaris Cluster information products.
3	Verify Sun QFS or SAM-QFS installation requirements.	Additional requirements for installing Sun QFS or SAM-QFS software with Solaris Cluster software are basically the same as for Sun QFS and SAM-QFS software without Solaris Cluster software. See Preparing for Installation.
4	Install Sun QFS or SAM-QFS software.	See Installing and Configuring SAM-QFS or Installing Sun QFS, as applicable.
5	(Optional) Configure failover on the metadata server for clustered (shared) file systems.	See Configuring Sun QFS Shared File Systems With Solaris Cluster.
6	(Optional) Configure high availability for archiving file systems (HA-SAM).	See Configuring SAM-QFS Archiving in a Solaris Cluster Environment (HA-SAM).
7	(Optional) Configure failover for unshared, unarchiving file systems (HA-QFS).	See Configuring Sun QFS Failover File Systems With Solaris Cluster.
8	(Optional) Configure clients that are part of the shared file system but that are not part of the Solaris Cluster configuration.	See Configuring Clients Outside of the Cluster.

See Also

- Requirements for Using SAM-QFS With Solaris Cluster
- Installing Sun QFS
- Installing and Configuring SAM-QFS
- Configuring Sun QFS Shared File Systems With Solaris Cluster
- Configuring Sun QFS Failover File Systems With Solaris Cluster
- Configuring SAM-QFS Archiving in a Sun Cluster Environment
- Configuring Clients Outside of the Cluster

Adding and Removing Nodes

Contents

Adding and Removing Nodes in an Existing Solaris Cluster Configuration

How to Add a Shared File System to an Existing Cluster

1. On a running cluster, prepare the cluster for adding a node.
   For detailed information, see http://docs.sun.com/app/docs/doc/820-2558/cacjggea?l=en&a=view. A summary of the steps follows:
   a. Become superuser and execute the clsetup utility.

   ```
   # clsetup
   ```

   b. Add the new nodes by name.
   c. Answer the questions when prompted.
   d. Verify that the command completed successfully.
   e. Choose the option to ignore all requests over the public network from any new machine trying to add itself to the cluster.
   f. Quit the `clsetup` utility.
2. Install the Solaris Cluster software on the new node.
   For detailed information, see http://docs.sun.com/app/docs/doc/820-4677/z40001fb1003552.

   > Note
   > You must install the same Solaris Cluster version on all nodes.

3. Once the new node is part of the cluster, install Sun QFS or SAM-QFS packages and reconfigure the `/etc/opt/SUNWsamfs/hosts.` file-system, on all nodes.
   For more information, see How to Add a Client Host to a Shared File System.

   > Note -
   > Be sure you configure the `/etc/opt/SUNWsamfs/hosts.`file-system with `hostname clusternodeX-priv "MDS Priority"` or add as a client.
   > In a shared file system configuration with Solaris Cluster, the `/etc/opt/SUNWsamfs/hosts.`file-system configuration files must contain Solaris Cluster private host names, and Solaris Cluster private host name IP addresses.

4. Mount the shared file systems.
5. Add the host to the resource group.
   Use the `clresourcegroup add-node` command and add the node to the end of the node list. For more information about this command, see the `clrg(1CL)` man page.

Example – Adding a Node to the Nodelist Property

The following command adds the node `phys-schost-4` to the Nodelist property for all resource groups.

```
# clresourcegroup set -p Nodelist+=phys-schost-4 +
```

How to Remove a Shared File System From a Cluster

To remove a node from a cluster, follow these instructions in the Solaris Cluster documentation.

To remove a host from a shared file system, see How to Remove a Client Host From a Shared File System.

Related Topics

- Using SAM-QFS With Solaris Cluster
- Configuring Archiving in a Solaris Cluster Environment (HA-SAM)

cldevice Example

Example – Verifying Devices and Device Redundancy

This example shows how to use output from the `cldevice`(1CL) command to find the devices in the Solaris Cluster environment, determine which devices are highly available, and then determine which devices are redundant.

Determining High Availability

The following example shows the `cldevice show` Solaris Cluster command to list paths of the devices in the DID configuration file for all nodes. In the output from the `cldevice show` command, look for output that shows a device that is visible from two or more nodes and that has the same World Wide Name. These are global devices.

The example uses Sun StorageTek T3 arrays in a RAID-5 configuration. The output shows that you can use devices 3 and 4 for configuring the disk cache for a file system.

Example - **`cldevice`** Command Example

```
ash# cldevice show | grep Device

+=== DID Device Instances ===
DID Device Name:                                /dev/did/rdsk/d1
  Full Device Path:                               ash:/dev/rdsk/c0t0d0
DID Device Name:                                /dev/did/rdsk/d2
  Full Device Path:                               ash:/dev/rdsk/c0t6d0
DID Device Name:                                /dev/did/rdsk/d3
  Full Device Path:                               ash:/dev/rdsk/c6t50020F2300004921d1
  Full Device Path:                               elm:/dev/rdsk/c6t50020F2300004921d1
DID Device Name:                                /dev/did/rdsk/d4
  Full Device Path:                               ash:/dev/rdsk/c6t50020F2300004921d0
  Full Device Path:                               elm:/dev/rdsk/c6t50020F2300004921d0
...

     _# The preceding output indicates that both ash and elm can access DID devices {{d3}} and
{{d4}}._
     _# These disks are highly available._


ash# format /dev/did/rdsk/d4s2
selecting /dev/did/rdsk/d4s2
[disk formatted]


FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        repair     - repair a defective sector
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        inquiry    - show vendor, product and revision
        volname    - set 8-character volume name

        <cmd>     - execute <cmd>, then return
        quit
format> verify

Primary label contents:

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 192 sec 64>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =  192
nsect       =   64
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264      101.16GB    (17265/0/0) 212152320
  1        usr    wm   17265 - 34529      101.16GB    (17265/0/0) 212152320
  2     backup    wu       0 - 34529      202.32GB    (34530/0/0) 424304640
  3 unassigned    wu       0                0         (0/0/0)             0
  4 unassigned    wu       0                0         (0/0/0)             0
  5 unassigned    wu       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7 unassigned    wu       0                0         (0/0/0)             0
```

Analyzing the Output From the Commands

The `cldevice show` command in this example lists device `/dev/rdsk/c6t50020F2300004921d0`, which is DID device `/dev/did/rdsk/d4` or global device `/dev/global/rdsk/d4`. This device has two partitions (0 and 1), each of which yields 212152320 blocks for use by a Sun QFS highly available local file system as `/dev/global/rdsk/d4s0` and `/dev/global/rdsk/d4s1`.

You need to issue the `cldevice show` and `format` commands for all devices to be configured for use by the Sun QFS highly available local file system.

- If you want to configure a Sun QFS shared file system on a cluster, you must use highly available, redundant devices.
- If you want to configure a Sun QFS highly available local file system and the `cldevice show` command output indicates that the devices you want to use are JBOD (just a bunch of disks) or dual-port SCSI disk devices, you need to use a volume manager that is supported in a Solaris Cluster environment to obtain redundancy. The options available and capabilities provided by such a volume manager are beyond the scope of this manual.

For more information about configuring devices that are on redundant storage, see your Solaris Cluster software installation documentation.

Configuring Archiving in a Solaris Cluster Environment (HA-SAM)

Contents

Configuring SAM-QFS Archiving in a Solaris Cluster Environment (HA-SAM)

About HA-SAM

High availability Sun Storage Archive Manager (HA-SAM) is an interface between a Sun QFS file system and Solaris Cluster software running on Solaris for SPARC and x64 hardware. The HA-SAM Solaris Cluster agent periodically monitors the health of SAM-QFS archiving operations on the primary node. In the event of an unrecoverable problem, the agent switches the SAM-QFS archiving and staging operations to a healthy node. Both voluntary and involuntary failover are supported on active-passive configurations. Only two-node active-passive configurations are supported.

For tape archiving and staging to continue after failover, tape drives must be visible to all nodes in a cluster on which HA-SAM is running, but they should not be configured as SAM-QFS shared drives. HA-SAM also supports disk archiving with disk archives visible to all nodes in a cluster.

HA-SAM depends on the Sun QFS Solaris Cluster agent and assumes that the Sun QFS file systems are mounted and managed by the Sun QFS agent. HA-SAM requires that the SAM-QFS catalog and stager directories be linked from the standard location to a directory in an HAStoragePlus file system. HA-SAM should be a resource in a resource group that contains Sun QFS and catalog resources.

In order to configure the HA-SAM Solaris Cluster agent, the Sun QFS Solaris Cluster agent must already be configured. These instructions assume that you have QFS configured in a Solaris Cluster environment using a shared QFS file system as described in Configuring Sun QFS Shared File Systems With Solaris Cluster.

Before You Begin

The following are requirements and restrictions that you should be aware of before configuring this feature.

- The nodes on which HA-SAM will be configured must be running the same version of Solaris (10) and Solaris Cluster (3.2 U2 or newer). Mixed versions of Solaris or Solaris Cluster software are not supported. No operating system other than Solaris is supported.
- The nodes in the cluster running HA-SAM must have the same type of architecture: SPARC or x64. Mixed architectures are not supported.
- No more than two nodes can be configured.
- Only active-passive configurations are supported, not active-active.
- Active I/O to HA-SAM file systems is supported only on the active node of an HA-SAM file system.
- Only shared QFS file systems are supported. Both `ma`- and `ms`-type file systems are supported. Stand-alone QFS environments are not supported.
- No software volume managers are supported with this configuration.
- Within the HA-SAM environment, the `nosam` mount option must be specified for any non-HA-SAM QFS file systems.
- The HA-SAM resource, QFS file systems, and HAStoragePlus file systems must all be configured within the same resource group. A separate Solaris Cluster resource group must be created for non-HA-SAM file systems.
- When using the `SUNW.hasam` resource type, you cannot specify the `bg` mount option in the `/etc/vfstab` file.
- Fibre tape drives are required. Tape drives must be visible to all systems through the fibre fabric, but should not be configured as

SAM-QFS shared drives.

- Disk volumes for disk archiving must be visible to all nodes.
- The active metadata server and potential metadata server must not be configured as a SAM-Remote client or server.
- The SAM-QFS catalog and stager directory must be in the default location: `/var/opt/SUNWsamfs/`. If it is any other location, the cluster nodes will not be able to locate it.
- Before configuring HA-SAM, verify that all SAM-QFS archiving operations are working correctly on the required nodes in the cluster.
- Only highly available (HA) agents are supported; no scalable agents are supported.
- Oracle software is not supported with this configuration, but the HA-SAM agent is supported as an RMAN target for Oracle backups.
- Voluntary failover forces a failover after five minutes in order to avoid problems with other potential HA agents.
- During voluntary failover, active archiving and staging times-out and terminates after less than 5 minutes.
- During failover, all SAM-QFS file systems must be failed over in order to avoid complications with the recycler that could result in loss of data.
- Sun StorageTek network attached tape libraries are supported, but not ADIC or Sony 8400 PetaSite Series automated tape libraries.
- HA-SAM environments cannot be managed by the SAM-QFS Manager browser interface.
- During involuntary failover, error messages for idled tape drives may be seen in the `/var/adm/messages` log file. These messages can safely be ignored. They simply indicate that HA-SAM did not have time to idle the drives before failover.
- After failover, volumes in the catalog may be marked with the E flag. This can occur when the software is unable to write the correct label at the end of a tape. To clear this error, see How to Clear Media Errors.

How to Confirm the Catalog and Stager Symbolic Links

In order to configure for high availability, the SAM-QFS catalog and stager queues must be in an HAStoragePlus file system. During the software installation process symbolic links should have been created from `/var/opt/SUNWsamfs` to the shared HAStoragePlus file system. This must be confirmed before continuing with the configuration process.

1. Verify that the `/var/opt/SUNWsamfs/catalog` and `/var/opt/SUNWsamfs/stager` contain a symbolic link to the HAStoragePlus mount point.
2. View the catalog and stager files.

```
# ls -l /var/opt/SUNWsamfs/catalog /var/opt/SUNWsamfs/stager
```

3. Verify that the output shows symbolic links similar to the following example.

```
lrwxrwxrwx 1 root other 19 Sep 30 11:05
/var/opt/SUNWsamfs/catalog -> /sam_shared/catalog
lrwxrwxrwx 1 root other 18 Sep 30 11:05
/var/opt/SUNWsamfs/stager -> /sam_shared/stager
```

   If the symbolic links exist, proceed to How to Configure SAM-QFS Archiving in a Sun Cluster Environment. If the links do not exist, proceed to the next step.

4. Copy any existing SAM-QFS catalog information to a temporary location.
5. Delete the existing `catalog` and `stager` files from their current location.

```
# rm -rf /var/opt/SUNWsamfs/catalog /var/opt/SUNWsamfs/stager
```

6. Create the `/var/opt/SUNWsamfs/catalog` and `/var/opt/SUNWsamfs/stager` symbolic links to the HAStoragePlus mount point.
   The following example is for an HAStoragePlus file system mounted at `/sam_shared`.

```
# ln -s /sam_shared/catalog /var/opt/SUNWsamfs/catalog
# ln -s /sam_shared/stager /var/opt/SUNWsamfs/stager
```

7. Copy the catalog information saved in Step c to the new shared catalog.

```
# cp -rp /var/tmp/catalog/* /var/opt/SUNWsamfs/catalog
```

How to Configure SAM-QFS Archiving in a Solaris Cluster Environment

In this example procedure, two file systems are configured: `qfs1` and `qfs2`.

1. Create an `HA-SAM` resource group.

```
# clresourcegroup create -n scnode-A,scnode-B SAM-HA
```

2. Create and configure the `SUNW.qfs` resource.

```
# clresource create -g SAM-HA -t SUNW.qfs \
-x QFSFileSystem=/global/qfs1,/global/qfs2 qfs-res
```

3. Create and configure a SAM-QFS shared resource.

```
# clresource create -g SAM-HA -t SUNW.HAStoragePlus \
-x FilesystemMountPoints=/sam_shared -x AffinityOn=TRUE sam-hastp
```

4. Create the `HA-SAM` resource type.

```
# clresourcetype register SUNW.hasam
```

5. Create and configure the `SUNW.hasam` resource.

```
# clresource create -g SAM-HA -t SUNW.hasam -x QFSName=qfs1,qfs2 \
-x CatalogFileSystem=/sam_shared sam-ha
```

6. Create dependencies between resources within the resource group.

```
# clresource set -p Resource_dependencies=sam-hastp qfs-ha
# clresource set -p Resource_dependencies=qfs-ha sam-ha
```

7. Bring the resource group online.

```
# clresourcegroup manage SAM-HA
# clresourcegroup online -eM SAM-HA
```

Note
The QFS file systems must be mounted before bringing the resource group online.

8. Ensure that the resource group is functional on all configured nodes.

```
# clresourcegroup switch -n scnode-B SAM-HA
# clresourcegroup switch -n scnode-A SAM-HA
```

Using the **`samd hastop`** Command

The `samd` command has an option that is specifically for use with HA-SAM. The `samd hastop` command stops the archiver and stager daemons before stopping `sam-amld` and its children.

The `samd hastop` command must not be used by an administrator in either a stand-alone or shared SAM-QFS environment. In addition, the normal `samd stop` command must not be used on nodes under HA-SAM control.

Using Sun StorageTek Libraries with HA-SAM

If you are using Sun StorageTek libraries in an HA-SAM environment, the STK daemons need to be notified to use a forced dismount instead of a normal dismount in the case of an involuntary failover. To accomplish this, the `/var/run/hasam_running` file is created on the active Solaris Cluster node when the HA-SAM resource is brought online. This is a zero-byte file with `root` permissions. If this file is detected on a node, `sam-stkd` and `sam-stk_helper` use `force_media` to dismount the tape on the drive. This allows the successful initialization of the library and the drives after an involuntary failover. The `hasam_running` file is deleted when a SAM-QFS resource is brought offline on a node.

In order to ensure the correct failover behavior for Sun StorageTek libraries, make sure the `/var/run/hasam_running` is not deleted if found on a Solaris Cluster node that has an HA-SAM resource online.

Configuring Sun QFS Failover File Systems With Solaris Cluster

Contents

Configuring Sun QFS Local Failover File Systems With Solaris Cluster

This section contains instructions for configuring the Sun QFS software in a Solaris Cluster environment. Before carrying out the configuration procedures in this section, you must have installed the Sun QFS software as described in Installing Sun QFS.

> Note -
> The Oracle Real Application Clusters (RAC) application is the only scalable application that the Sun QFS software supports in Solaris Cluster environments. For more information, see the Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS. See also Troubleshooting Issues With Oracle RAC.

The Sun QFS file system can work with Solaris Cluster in one of the following ways:

- As a highly-available shared file system, simply referred to as a clustered file system. For information about configuring a clustered file system, see Configuring Sun QFS Shared File Systems With Solaris Cluster
- As a highly-available local file system, also known as a failover file system.

Task Map: Configuring Local Failover File Systems With Solaris Cluster

Step	Task	Description
1	Prepare the host systems.	Verify user and group IDs, and NFS-share the file system as described in Preparing the Host Systems.
2	Edit the `mcf` file.	Edit the `mcf` file for each host that you want to include in the Solaris Cluster environment. See Editing mcf Files for a Local Failover File System.
3	Make the highly-available file system available to Solaris Cluster.	Configure the resource as described in How to Configure a Highly Available File System as a SUNW.HAStoragePlus Resource.
4	Verify the resource group on all nodes.	From any node in the cluster, move the highly-available resource to another node as described in How to Verify the Resource Group on All Nodes.

Preparing the Host Systems

To prepare the host systems for a shared file system in a Solaris Cluster environment, perform the following tasks:

1. Verify that all the hosts have the same user and group IDs.
   - If you are not running the Network Information Name service (NIS), make sure that all `/etc/passwd` and all `/etc/group` files are identical.
   - If you are running NIS, the `/etc/passwd` and `/etc/group` files should already be identical.
     For more information, see the `nis+`(1) man page.
2. NFS-share the file system.
   The following is a general description of how to NFS-share a file system in a Solaris Cluster environment. For more information on NFS-sharing file systems that are controlled by HAStoragePlus, see the Sun Cluster Data Service for Network File System (NFS) Guide for Solaris OS, and your NFS documentation.
   a. Locate the `dfstab.`resource-name file.
      The `Pathprefix` property of HAStoragePlus specifies the directory in which the `dfstab.`resource-name file resides.
   b. Use `vi`(1) or another editor to add a `share`(1M) command to the Pathprefix/`SUNW.nfs/dfstab.`resource-name file.
      For example:

```
share -F nfs -o rw /global/qfs1
```

Editing **mcf** Files for a Local Failover File System

The lines that define a particular file system must be identical in the `mcf` files on all host systems that support the file system. Only one `mcf` file can reside on a host. Because you can have other, additional file systems defined in an `mcf` file, the `mcf` files on different hosts might not be identical.

> Note -
> If you update a metadata server's `mcf` file at any time after the shared file system is mounted, you must also update the `mcf` files as necessary on all hosts that can access that shared file system.

How to Edit mcf Files for a Failover File System

Perform this procedure to enable a local file system to failover in a Solaris Cluster environment. Perform these steps on each host that supports the file system.

1. Log in to the Solaris Cluster node.
2. Become superuser.
3. Use `vi`(1) or another editor to create an `mcf` file on that node.
   If an `mcf` file already exists on the host, add the lines for the new file system to this `mcf` file.
4. Copy the lines that define the file system from the primary node's `mcf` file to this node's `mcf` file.

Examples

Example 1 – Solaris OS Hosts

This example shows how the `samfsconfig`(1M) command is used to retrieve device information for family set `sharefs1` on client `tethys`. Because `tethys` is a potential metadata server, it is connected to the same metadata disks as `titan`.

Example – **samfsconfig**(1M) Command Example on **tethys**

```
tethys# samfsconfig /dev/dsk/*
#
# Family Set "sharefs1" Created Wed Jun 27 19:33:50 2003
#
sharefs1 10 ma sharefs1 on shared
/dev/dsk/c2t50020F23000065EEd0s6 11 mm sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12 mr sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13 mr sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14 mr sharefs1 on
```

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

- Each Device State field is set to on.
- The shared keyword appears in the Additional Parameters field for the file system name.
  The following example shows the resulting mcf file.

Example – **mcf** File for **sharefs1** Client Host **tethys**

```
# Equipment Eq Eq Family Dev Add
# Identifier Ord Type Set State Params
# ---------- --- ---- ------ ----- ------
sharefs1 10 ma sharefs1 on shared
/dev/dsk/c2t50020F23000065EEd0s6 11 mm sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12 mr sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13 mr sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14 mr sharefs1 on
```

The Equipment Number numbers match those of the mcf file for metadata server titan (see Configuration Example for a Shared File System on a Solaris OS Platform). These Equipment Number numbers must not already be in use on client host tethys or any other client host.

Example 2 – Solaris OS Hosts

This example shows how the samfsconfig(1M) command is used to retrieve device information for family set sharefs1 on client host mimas. Because mimas can never become a metadata server, it is not connected to the metadata disks.

Example – **samfsconfig**(1M) Command Example on **mimas**

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

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

- Create a file system Family Set declaration line, beginning with sharefs1, in the mcf file. Enter the shared keyword in the Additional Parameters field of the file system Family Set declaration line.
- Create one or more nodev lines for each missing Equipment Number field. For these lines, the keyword nodev must appear in the Equipment Identifier field for each inaccessible device. In this example, you create a device entry in the mcf file named nodev to represent the missing metadata disk.
- Ensure that each Device State field is set to on.
- Uncomment the device lines.
  The following example shows the resulting mcf file.

Example – **mcf** File for Client Host **mimas**

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

How to Configure a Failover File System as a SUNW.HAStoragePlus Resource

Perform this task if you are configuring a local file system to failover on a Solaris Cluster platform.

1. Use the `scrgadm`(1M) command to set the `FilesystemCheckCommand` property of HAStoragePlus to `/bin/true`.
   All other resource properties for HAStoragePlus apply as specified in `SUNW.HAStoragePlus`(5).
   The following example command shows how to use the `scrgadm`(1M) command to configure a `SUNW.HAStoragePlus` resource:

```
# scrgadm -a -g qfs-rg -j ha-qfs -t SUNW.HAStoragePlus \
        -x FilesystemMountPoints=/global/qfs1 \
        -x FilesystemCheckCommand=/bin/true
```

How to Verify the Resource Group on All Nodes

This task ensures that the file system can move from node to node when the Solaris Cluster software performs a failover.

Perform these steps for each node in the cluster, with a final return to the original server.

Steps

1. From any node in the Solaris Cluster environment, use the `scswitch`(1M) command to move the file system resource from one node to another.
   For example:

```
server# scswitch -z -g qfs-rg -h elm
```

2. Use the `scstat`(1M) command to verify that the file system resource was moved successfully.
   For example:

```
server# scstat
-- Resources --
Resource Name     Node Name  State     Status Message
-------------     ---------  -----     --------------
Resource: qfs-res   ash      Offline   Offline
Resource: qfs-res   elm      Online    Online
Resource: qfs-res   oak      Offline   Offline
```

Related Topics

- Adding and Removing Nodes
- Using SAM-QFS With Solaris Cluster
- Configuring Archiving in a Solaris Cluster Environment (HA-SAM)
- Troubleshooting Issues With Oracle RAC

Configuring Sun QFS Shared File Systems With Solaris Cluster

Contents

- Related Topics

Configuring Sun QFS Shared File Systems With Solaris Cluster

This section contains instructions for configuring the Sun QFS software in a Solaris Cluster environment. Before carrying out the configuration procedures in this section, you must have installed the Sun QFS software as described in Installing Sun QFS.

Note -

- You can also have clients that are part of the shared file system but that are not part of the Solaris Cluster configuration. For more information, see Configuring Clients Outside of the Cluster.
- The Oracle Real Application Clusters (RAC) application is the only scalable application that the Sun QFS software supports in Solaris Cluster environments. For more information, see the Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS. See also Troubleshooting Issues With Oracle RAC.

The Sun QFS file system can work with Solaris Cluster in one of the following ways:

- As a highly-available shared file system, simply referred to as a clustered file system.
- As a highly-available local file system, also known as a failover file system. For information about configuring a failover file system, see Configuring Sun QFS Failover File Systems With Solaris Cluster.

You can also add a shared file system to an existing Solaris Cluster environment. For information, see Adding and Removing Nodes.

Task Map: Configuring Clustered File Systems With Solaris Cluster

Step	Task	Description
1	Prepare the host systems.	Verify user and group IDs, and NFS-share the file system as described in Preparing the Host Systems.
2	Edit the `mcf` file.	Edit the `mcf` file for each host that you want to include in a shared file system in a Solaris Cluster environment. See Editing mcf Files for a Clustered File System.
3	Create the shared hosts file.	On the metadata server, create the shared hosts file as described in Creating the Shared Hosts File on the Metadata Server.
4	Verify that the shared file system is mounted.	On each host that can mount the file system, verify that the file system is mounted and shared. See How to Verify the Daemons.
5	Make the shared file system available to Solaris Cluster.	Log in to the metadata server and configure the resource as described in How to Enable a Shared File System as a SUNW.qfs(5) Resource.
6	Bring the shared resource online.	On the metadata server, move the shared resource to another node as described in How to Bring the Shared Resource Online.
7	Verify the resource group on all nodes.	On each node in the cluster that is part of the shared file system, move the shared resource to another node as described in How to Verify the Resource Group on All Nodes. When you are finished, return to the metadata server.

Preparing the Host Systems

To prepare the host systems for a shared file system in a Solaris Cluster environment, perform the following tasks:

1. Verify that all the hosts have the same user and group IDs.
   - If you are not running the Network Information Name service (NIS), make sure that all `/etc/passwd` and all `/etc/group` files are identical.
   - If you are running NIS, the `/etc/passwd` and `/etc/group` files should already be identical.
     For more information, see the `nis+`(1) man page.
2. NFS-share the file system.
   The following is a general description of how to NFS-share a file system in a Solaris Cluster environment. For more information on

NFS-sharing file systems that are controlled by HAStoragePlus, see the Sun Cluster Data Service for Network File System (NFS) Guide for Solaris OS, and your NFS documentation.

a. Locate the `dfstab.`resource-name file.
   The `Pathprefix` property of HAStoragePlus specifies the directory in which the `dfstab.`resource-name file resides.
b. Use `vi`(1) or another editor to add a `share`(1M) command to the Pathprefix/`SUNW.nfs/dfstab.`resource-name file.
   For example:

```
share -F nfs -o rw /global/qfs1
```

Editing **mcf** Files for a Clustered File System

The lines that define a particular file system must be identical in the `mcf` files on all host systems that support the file system. Only one `mcf` file can reside on a host. Because you can have other, additional file systems defined in an `mcf` file, the `mcf` files on different hosts might not be identical.

> Note -
> If you update a metadata server's `mcf` file at any time after the shared file system is mounted, you must also update the `mcf` files as necessary on all hosts that can access that shared file system.

How to Edit mcf Files for a Clustered File System

Perform this procedure for each host that you want to include in a shared file system in a Solaris Cluster environment.

1. Log in to the host.
2. Become superuser.
3. Use the `format`(1M) command to verify the presence of client host disks.
4. Use `vi`(1) or another editor to create an `mcf` file.
   If an `mcf` file already exists on the host, add the lines for the new file system to this `mcf` file.
5. Use the `samfsconfig`(1M) command to locate the local device names for each additional host to be configured in the shared file system.
   The `samfsconfig`(1M) command generates configuration information that can help you to identify the devices included in the shared file system. Enter a separate `samfsconfig`(1M) command on each client host. Note that the controller number might not be the same controller number as on the metadata server because the controller numbers are assigned by each client host.
6. Update the `mcf` file on other client hosts.
   To access or mount a shared file system, a host system must have that file system defined in its `mcf` file. The content of the `mcf` files varies, depending on the type of Solaris Cluster hosts:
   - The primary metadata server
   - Potential metadata servers
     All hosts can be metadata servers because the Solaris Cluster software fails over system resources in the event of a node failure.
     Use `vi`(1) or another editor to edit the `mcf` file on one of the client host systems. The `mcf` file must be updated on all client hosts to be included in the shared file system. The file system and disk declaration information must have the same data for the Family Set name, Equipment Number, and Equipment Type fields as the configuration on the metadata server. The `mcf` files on the client hosts must also include the `shared` keyword. The device names, however, can change as controller assignments can change from host to host.

Example 1 – **mcf** File Using Hardware RAID and Solaris Cluster

The following sample `mcf` file includes four shared file systems. The first three are on Solaris Cluster nodes. The last file system is not part of the cluster.

```
psfb1% more /etc/opt/SUNWsamfs/mcf
OraHomes 20 ma OraHomes - shared
/dev/did/dsk/d1s0 21 mm OraHomes -
/dev/did/dsk/d2s0 22 mr OraHomes -
/dev/did/dsk/d3s0 23 mr OraHomes -

CrsHomes 30 ma CrsHomes - shared
/dev/did/dsk/d6s0 31 mm CrsHomes -
/dev/did/dsk/d7s0 32 mr CrsHomes -

OraData 40 ma OraData - shared
/dev/did/dsk/d8s0 41 mm OraData -
/dev/did/dsk/d9s0 42 mr OraData -
/dev/did/dsk/d10s0 43 mr OraData -
/dev/did/dsk/d11s0 44 mr OraData -

votingdisk1 50 ms votingdisk1 - shared
/dev/did/dsk/d4s1 51 md votingdisk1 -
votingdisk2 60 ms votingdisk2 - shared
/dev/did/dsk/d4s2 61 md votingdisk2 -
votingdisk3 70 ms votingdisk3 - shared
/dev/did/dsk/d4s3 71 md votingdisk3 -
```

Creating the Shared Hosts File on the Metadata Server

The shared hosts configuration file must reside in the following location on the metadata server:

```
/etc/opt/SUNWsamfs/hosts.<fsname>
```

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

The following table shows the fields in the hosts configuration file.

Table – Shared Hosts Configuration File Fields

Field	Content
Host Name	This field must contain the alphanumeric name of a metadata server or potential metadata server that is part of the Sun QFS shared file system.
Host Interfaces	This field must contain a comma-separated list of host interface addresses. This field can be created from the output received from the `ifconfig`(1M) `-a` command. The individual interfaces can be specified in one of the following ways: * Dotted-decimal IP address form * IP version 6 hexadecimal address form * As a symbolic name that the local domain name service (DNS) can resolve to a particular host interface Each host uses this field to determine whether it will try to connect to the specified host interface. The system evaluates the addresses from left to right, and the connection is made using the first responding address in the list that is also included in the shared hosts file.

In a shared file system, each client host obtains the list of metadata server IP addresses from the metadata server host.

The metadata server and the client hosts use both the `/etc/opt/SUNWsamfs/hosts.`fsname file on the metadata server and the `hosts.`fsname`.local` file on each client host (if it exists) to determine the host interface to use when accessing the file system.

How Metadata Server Addresses Are Obtained

The information in this section might be useful when you are debugging.

In a shared file system, each client host obtains the list of metadata server IP addresses from the shared hosts file.
The metadata server and the client hosts use the shared hosts file on the metadata server and the `hosts.`fsname`.local` file on each client host (if it exists) to determine the host interface to use when accessing the metadata server. This process is as follows:

> Note -
> The term client, as in network client, is used to refer to both client hosts and the metadata server host.

1. The client obtains the list of metadata server host IP interfaces from the file system's on-disk shared hosts file. To examine this file, issue the `samsharefs`(1M) command from the metadata server or from a potential metadata server.
2. The client searches for an `/etc/opt/SUNWsamfs/hosts.`fsname`.local` file. Depending on the outcome of the search, one of the following occurs:
    - If a `hosts.`fsname`.local` file does not exist, the client attempts to connect, in turn, to each address in the server's line in the shared hosts file until it succeeds in connecting.
    - If the `hosts.`fsname`.local` file exists, the client performs the following tasks:
3. It compares the list of addresses for the metadata server from both the shared hosts file on the file system and the `hosts.`fsname `.local` file.
4. It builds a list of addresses that are present in both places, and then it attempts to connect to each of these addresses, in turn, until it succeeds in connecting to the server. If the order of the addresses differs in these files, the client uses the ordering in the `hosts.` fsname`.local` file.

Example – Sun QFS Shared File System Hosts File Example

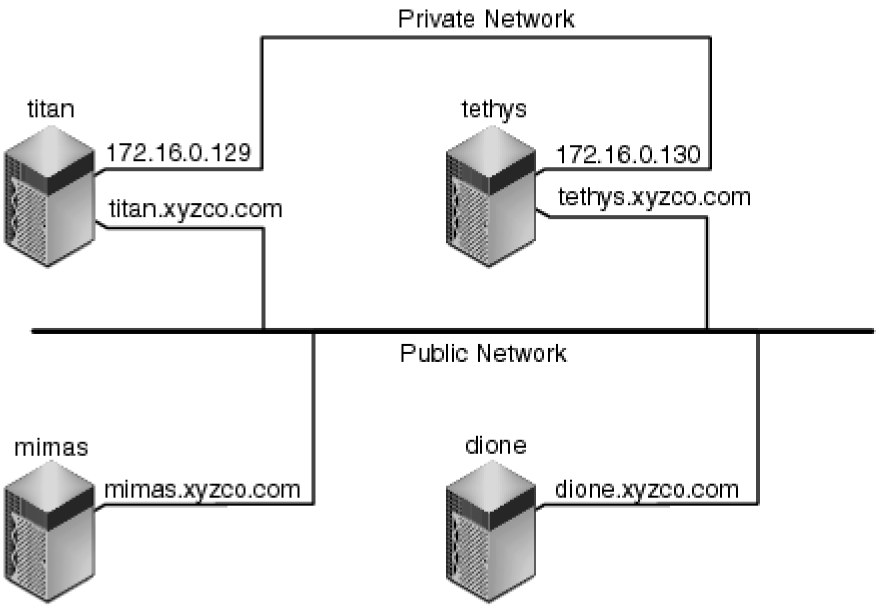
This set of examples shows a detailed scenario for a shared file system that comprises four hosts.

The following example shows a hosts file that lists four hosts.

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IP Server Not Server
# Name Addresses Priority Used Host
# ---- ----------------- -------- ---- -----
titan 172.16.0.129 1 - server
tethys 172.16.0.130 2 -
mimas mimas - -
dione dione - -
```

The following figure shows the interfaces to these systems.

Figure – Network Interfaces for Sun QFS Shared File System Hosts File Example



Example – File **`hosts.sharefs1.local`** on **`titan`** and **`tethys`**

Systems `titan` and `tethys` share a private network connection with interfaces `172.16.0.129` and `172.16.0.130`. To guarantee that `titan` and `tethys` always communicate over their private network connection, the system administrator has created identical copies of `/etc/opt/SUNWsamfs/hosts.sharefs1.local` on each system.

The following example shows the information in the `hosts.sharefs1.local` files on `titan` and `tethys`.

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

Example – File **`hosts.sharefs1.local`** on **`mimas`** and **`dione`**

Systems `mimas` and `dione` are not on the private network. To guarantee that they always connect to `titan` and `tethys` through `titan`'s and `tethys`'s public interfaces, the system administrator has created identical copies of `/etc/opt/SUNWsamfs/hosts.sharefs1.local` on `mimas` and `dione`.

The following example shows the information in the `hosts.sharefs1.local` files on `mimas` and `dione`.

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

How to Verify the Daemons

Perform this task on each host that can mount the file system.

1. Verify that the file system is mounted.
   If it is not mounted, go back to Mounting the File System and follow the instructions there.
2. Use the `ps`(1) and `grep`(1) commands to determine whether the `sam-sharefsd` daemon is running for this file system.
   For example:

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

   This example shows that the `sam-sharefsd` daemon is active for the `sharefs1` file system.

   > Note -
   > If the `sam-sharefsd` daemon is active for your shared file system, you need to perform some diagnostic procedures. For information about these procedures, see Using Daemons, Processes, and Tracing.

3. If the output from this command indicates that the `sam-sharefsd` daemon is not running, determine whether the `sam-fsd` daemon is running as follows:
   a. Use the `ps`(1) and `grep`(1) commands to verify that the `sam-fsd` daemon is running for this file system.
   b. Examine the output.
      The following code example shows `sam-fsd` output that indicates that the daemon is running.

      ```
      cur% ps -ef | grep sam-fsd
      user1 16435 16314  0 16:52:36 pts/13   0:00 grep sam-fsd
      root    679     1  0   Aug 24 ?        0:00 /usr/lib/fs/samfs/sam-fsd
      ```

   c. Do one of the following:
      - If the output indicates that the `sam-fsd` daemon is not running, and if no file system has been accessed since the system's last boot, issue the `samd`(1M) `config` command, as follows:

```
# samd config
```

- If the output indicates that the `sam-fsd` daemon is running, enable tracing in the `defaults.conf` file and check the following files to determine whether configuration errors are causing the problem:
  - `/var/opt/SUNWsamfs/trace/sam-fsd`
  - `/var/opt/SUNWsamfs/trace/sam-sharefsd`

How to Enable a Shared File System as a SUNW.qfs(5) Resource

1. Log in to the metadata server as superuser.
2. Use the `scrgadm`(1M) `-p` command to search for the `SUNW.qfs`(5) resource type.
   For example:

```
metadataserver# scrgadm -p | grep SUNW.qfs
```

3. If the `SUNW.qfs` resource type is missing, issue the following command:

```
metadataserver# scrgadm -a -t SUNW.qfs
```

4. Use the `scrgadm`(1M) command to set the `FilesystemCheckCommand` property of the `SUNW.qfs`(5) resource type to `/bin/true`
   .
   The `SUNW.qfs`(5) resource type is part of the Sun QFS software package. Configuring the resource type for use with your shared file system makes the shared file system's metadata server highly available. Solaris Cluster scalable applications can then access data contained in the file system.
   The following code example shows how to use the `scrgadm`(1M) command to register and configure the `SUNW.qfs` resource type. In this example, the nodes are `scnode-A` and `scnode-B`. `/global/sharefs1` is the mount point as specified in the `/etc/vfstab` file.

```
# scrgadm -a -g qfs-rg -h scnode-A,scnode-B
# scrgadm -a -g qfs-rg -t SUNW.qfs -j qfs-res \
       -x QFSFileSystem=/global/sharefs1
```

> Note -
> In a SAM-QFS environment, you can also configure the archiving features for high availability using Solaris Cluster software. For instructions, see Configuring Archiving in a Solaris Cluster Environment (HA-SAM).

How to Bring the Shared Resource Online

1. Verify that the file system is mounted on all nodes.

> Note -
> If you are using the `SUNW.qfs` resource type, you cannot use the `bg` mount option in the `/etc/vfstab` file.

2. Log in to the node upon which the file system is based.
3. Use the `scswitch`(1M) command to move the file system resource to another node.
   For example:

```
metadataserver# scswitch -Z -g qfs-rg
```

4. Use the `scstat`(1M) command to verify that the file system resource was moved successfully.
   For example:

```
metadataserver# scstat

<information deleted from this output>

-- Resources --
Resource Name    Node Name  State     Status Message
-------------    ---------  -----     --------------
Resource: qfs-res   ash     Online    Online
Resource: qfs-res   elm     Offline   Offline
Resource: qfs-res   oak     Offline   Offline
```

How to Verify the Resource Group on All Nodes

Perform these steps for each node in the cluster, with a final return to the metadata server.

Steps

1. From any node in the Solaris Cluster environment, use the `scswitch`(1M) command to move the file system resource from one node to another.
   For example:

```
server# scswitch -z -g qfs-rg -h elm
```

2. Use the `scstat`(1M) command to verify that the file system resource was moved successfully.
   For example:

```
server# scstat
-- Resources --
Resource Name    Node Name  State     Status Message
-------------    ---------  -----     --------------
Resource: qfs-res   ash     Offline   Offline
Resource: qfs-res   elm     Online    Online
Resource: qfs-res   oak     Offline   Offline
```

Related Topics

- Adding and Removing Nodes
- Using SAM-QFS With Solaris Cluster
- Configuring Archiving in a Solaris Cluster Environment (HA-SAM)
- Troubleshooting Issues With Oracle RAC

Editing mcf Files for a Clustered File System

Contents

Editing **`mcf`** Files for a Clustered File System

The lines that define a particular file system must be identical in the `mcf` files on all host systems that support the file system. Only one `mcf` file can reside on a host. Because you can have other, additional file systems defined in an `mcf` file, the `mcf` files on different hosts might not be identical.

> Note -
> If you update a metadata server's `mcf` file at any time after the shared file system is mounted, you must also update the `mcf` files as necessary on all hosts that can access that shared file system.

How to Edit mcf Files for a Clustered File System

Perform this procedure for each host that you want to include in a shared file system in a Solaris Cluster environment.

1. Log in to the host.
2. Become superuser.
3. Use the `format`(1M) command to verify the presence of client host disks.
4. Use `vi`(1) or another editor to create an `mcf` file.
   If an `mcf` file already exists on the host, add the lines for the new file system to this `mcf` file.
5. Use the `samfsconfig`(1M) command to locate the local device names for each additional host to be configured in the shared file system.
   The `samfsconfig`(1M) command generates configuration information that can help you to identify the devices included in the shared file system. Enter a separate `samfsconfig`(1M) command on each client host. Note that the controller number might not be the same controller number as on the metadata server because the controller numbers are assigned by each client host.
6. Update the `mcf` file on other client hosts.
   To access or mount a shared file system, a host system must have that file system defined in its `mcf` file. The content of the `mcf` files varies, depending on the type of Solaris Cluster hosts:
   - The primary metadata server
   - Potential metadata servers
     All hosts can be metadata servers because the Solaris Cluster software fails over system resources in the event of a node failure.
     Use `vi`(1) or another editor to edit the `mcf` file on one of the client host systems. The `mcf` file must be updated on all client hosts to be included in the shared file system. The file system and disk declaration information must have the same data for the Family Set name, Equipment Number, and Equipment Type fields as the configuration on the metadata server. The `mcf` files on the client hosts must also include the `shared` keyword. The device names, however, can change as controller assignments can change from host to host.

Example 1 – **`mcf`** File Using Hardware RAID and Solaris Cluster

The following sample `mcf` file includes four shared file systems. The first three are on Solaris Cluster nodes. The last file system is not part of the cluster.

```
psfb1% more /etc/opt/SUNWsamfs/mcf
OraHomes 20 ma OraHomes - shared
/dev/did/dsk/d1s0     21    mm   OraHomes  -
/dev/did/dsk/d2s0    22    mr   OraHomes  -
/dev/did/dsk/d3s0    23    mr   OraHomes  -

CrsHomes 30 ma CrsHomes - shared
/dev/did/dsk/d6s0    31    mm   CrsHomes  -
/dev/did/dsk/d7s0    32    mr   CrsHomes  -

OraData 40 ma OraData - shared
/dev/did/dsk/d8s0   41    mm   OraData  -
/dev/did/dsk/d9s0    42    mr   OraData  -
/dev/did/dsk/d10s0    43    mr   OraData  -
/dev/did/dsk/d11s0    44    mr   OraData  -

votingdisk1 50 ms votingdisk1 - shared
/dev/did/dsk/d4s1   51   md votingdisk1 -
votingdisk2 60 ms votingdisk2 - shared
/dev/did/dsk/d4s2   61   md votingdisk2 -
votingdisk3 70 ms votingdisk3 - shared
/dev/did/dsk/d4s3   71   md votingdisk3 -
```

Related Topics

- Editing mcf Files for a Local Failover File System
- Adding and Removing Nodes
- Using SAM-QFS With Solaris Cluster
- Configuring Archiving in a Solaris Cluster Environment (HA-SAM)
- Troubleshooting Issues With Oracle RAC

Editing mcf Files for a Local Failover File System

Contents

Editing **`mcf`** Files for a Local Failover File System

The lines that define a particular file system must be identical in the `mcf` files on all host systems that support the file system. Only one `mcf` file can reside on a host. Because you can have other, additional file systems defined in an `mcf` file, the `mcf` files on different hosts might not be identical.

> Note -
> If you update a metadata server's `mcf` file at any time after the shared file system is mounted, you must also update the `mcf` files as necessary on all hosts that can access that shared file system.

How to Edit mcf Files for a Failover File System

Perform this procedure to enable a local file system to failover in a Solaris Cluster environment. Perform these steps on each host that supports the file system.

1. Log in to the Solaris Cluster node.
2. Become superuser.
3. Use `vi`(1) or another editor to create an `mcf` file on that node.
   If an `mcf` file already exists on the host, add the lines for the new file system to this `mcf` file.
4. Copy the lines that define the file system from the primary node's `mcf` file to this node's `mcf` file.

Examples

Example 1 – Solaris OS Hosts

This example shows how the `samfsconfig`(1M) command is used to retrieve device information for family set `sharefs1` on client `tethys`. Because `tethys` is a potential metadata server, it is connected to the same metadata disks as `titan`.

Example – **`samfsconfig`**(1M) Command Example on **`tethys`**

```
tethys# samfsconfig /dev/dsk/*
#
# Family Set "sharefs1" Created Wed Jun 27 19:33:50 2003
#
sharefs1                         10 ma sharefs1 on shared
/dev/dsk/c2t50020F23000065EEd0s6 11 mm sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12 mr sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13 mr sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14 mr sharefs1 on
```

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

- Each Device State field is set to on.
- The shared keyword appears in the Additional Parameters field for the file system name.
  The following example shows the resulting mcf file.

Example – **mcf** File for **sharefs1** Client Host **tethys**

```
# Equipment                        Eq  Eq   Family   Dev   Add
# Identifier                       Ord Type Set      State Params
# ----------                       --- ---- ------   ----- ------
sharefs1                           10  ma   sharefs1 on    shared
/dev/dsk/c2t50020F23000065EEd0s6 11  mm   sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12  mr   sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13  mr   sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14  mr   sharefs1 on
```

The Equipment Number numbers match those of the mcf file for metadata server titan (see Configuration Example for a Shared File System on a Solaris OS Platform). These Equipment Number numbers must not already be in use on client host tethys or any other client host.

Example 2 – Solaris OS Hosts

This example shows how the samfsconfig(1M) command is used to retrieve device information for family set sharefs1 on client host mimas. Because mimas can never become a metadata server, it is not connected to the metadata disks.

Example – **samfsconfig**(1M) Command Example on **mimas**

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

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

- Create a file system Family Set declaration line, beginning with sharefs1, in the mcf file. Enter the shared keyword in the Additional Parameters field of the file system Family Set declaration line.
- Create one or more nodev lines for each missing Equipment Number field. For these lines, the keyword nodev must appear in the Equipment Identifier field for each inaccessible device. In this example, you create a device entry in the mcf file named nodev to represent the missing metadata disk.
- Ensure that each Device State field is set to on.
- Uncomment the device lines.
  The following example shows the resulting mcf file.

Example – **mcf** File for Client Host **mimas**

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

Related Topics

- Adding and Removing Nodes
- Using SAM-QFS With Solaris Cluster
- Configuring Archiving in a Solaris Cluster Environment (HA-SAM)
- Troubleshooting Issues With Oracle RAC

Requirements for Using SAM-QFS With Solaris Cluster

Contents

Requirements for Using SAM-QFS With Solaris Cluster

Consider the following requirements before you attempt to configure Solaris Cluster with Sun QFS or SAM-QFS.

Basic Product Familiarity

You must be familiar with basic Solaris Cluster concepts and operations.

In a Solaris Cluster environment, the disk space used by the file system must be configured on storage that is highly available and redundant. Ensure that you have a good understanding of the concepts in the Sun Cluster System Administration Guide for Solaris OS.
You should also be familiar with Solaris Cluster operations. For information on Solaris Cluster operations, see the following manuals:

- Sun Cluster Concepts Guide for Solaris OS
- Sun Cluster Software Installation Guide for Solaris OS
- Sun Cluster Data Services Planning and Administration Guide for Solaris OS

Hardware Requirements

You must have at least two UltraSPARC or AMD x64 hosts to use as a cluster.

> Note -
> You cannot mix hardware architectures in a shared Solaris Cluster environment. All of the nodes must be either SPARC or AMD x64.

Software Requirements

You must have the following minimum software levels installed on each cluster node:

- Solaris 10 5/08 OS
- Solaris Cluster 3.2 1/09

Each node must have the same Solaris Cluster software level and Solaris Cluster patch collection. You must install Sun QFS or SAM-QFS software packages on each node in the cluster that will host a Sun QFS file system.

Disk Device Requirements

For Clustered (Shared) File Systems

To configure a shared file system, use raw device identifier (DID) devices. In the `cldevice show` command output, these appear as `/dev/did/rdsk/dN` devices. The Solaris Cluster nodes that share the file system must have access to each DID device through a host bus adapter (HBA) direct connection. All devices must be accessible to the file system from all nodes in the Solaris Cluster environment that mount the shared file system. For more information about DID devices, see the `did`(7) man page.

When you specify these devices in your `mcf` file, use the `/dev/did/rdsk/dN` devices from the `cldevice show` command output. For more information about using the `cldevice`(1CL) command, see Example: Verifying Devices and Device Redundancy.

> Note -
> The Sun QFS software supports using multi-owner disk sets in Solaris Volume Manager for Solaris Cluster to obtain redundancy.

For Failover (Local) File Systems

To configure a local file system for failover, you must use highly available devices. You can use either of the following options:

- Raw devices - Use Solaris Cluster global devices. Use the output from the `cldevice show` command to determine the names of the global devices and substitute `global` for `did` when specifying the devices in the `mcf` file. Global devices are accessible from all nodes in a Solaris Cluster environment, even if these devices are not physically attached to all nodes. If all nodes that have a hardware connection to the disk lose their connections, the remaining nodes cannot access the disk. File systems created on global devices are not necessarily highly available.
- Volume manager - Use one of the following:
  - Solaris™ Volume Manager for Solaris Cluster for either shared or stand-alone Sun QFS configurations. Such devices are located in `/dev/md`.
  - Veritas Volume Manager (VxVM) for stand-alone QFS configurations only. Such devices are located in `/dev/vx`.

Use the `clsetup`(1CL) utility to register volume-managed devices with the Solaris Cluster framework before you configure your file system.

> Note -
> If you use a volume manager, use it only to provide redundancy. For performance reasons, do not use the volume manager to concatenate storage on separate devices; this causes the Sun QFS highly available file system to distribute I/O inefficiently across the component devices.

To confirm which devices in your Solaris Cluster environment are highly available, use the `cldevice show | grep Device` command . This command lists the paths of the devices in the DID configuration file. In the output from the `cldevice show` command, look for devices that have two or more DID devices listed with the identical DID device number. Such devices are highly available in a Solaris Cluster environment and can also be configured as global devices for a file system, even if they directly connect only to a single node.

I/O requests issued to global devices from a node other than the direct-attached node are issued over the Solaris Cluster interconnect. These single-node, global devices cease to be available when all nodes that have direct access to the device are unavailable.

Disk Device Redundancy

To implement redundancy, you have the following options:

- If you are configuring a shared file system, you can obtain redundancy through multi-owner disk sets in Solaris Volume Manager for the Solaris Cluster environment.
- If you are configuring a highly available file system, you can use either the Solaris Volume Manager or the Veritas Volume Manager to obtain mirroring (RAID-1) or striping (RAID-5).
  For more information about volume sizing and redundancy configurations, see the Solaris Volume Manager Administration Guide or your Veritas Volume Manager documentation.

Storage Redundancy

Storage redundancy is achieved by maintenance of extra disk copies of data using mirroring or RAID-1, or parity across several disks using RAID-5 to allow reconstruction of data after a disk failure. When supported by the hardware, these disk configurations enable you to configure the raw devices in a Solaris Cluster environment without a volume manager. These raw devices are accessible from multiple nodes, so you can issue the `format`(1M) command from any node to obtain information on the disks.
Storage redundancy can also be achieved by using software to support mirroring or RAID. This method, however, is not generally suitable for

concurrent access from multiple hosts. Solaris Cluster software supports mirroring of disk volumes (RAID-1 only) through its multi-owner diskset feature with Sun QFS software and Solaris Volume Manager. No other software redundancy is supported.

Data Path Redundancy

Data path redundancy is achieved with multiple HBAs, which are configured from a single node. If your environment includes multiple HBAs for redundancy, be aware that the Sun QFS file systems require multipathing software like the Solaris I/O multipathing feature (MPxIO) to enable data path redundancy. For more information, see the Solaris SAN Configuration and Multipathing Guide, or see the `stmsboot`(1M) man page.

To determine redundancy, consult the hardware documentation for your disk controllers and disk devices. You need to know whether the disk controller or disk devices that are reported by the `cldevice show` command are on redundant storage. For information, see the storage controller vendor's documentation set and view the current controller configuration.

Performance Considerations

For optimal file system performance, the metadata and file data should be accessible through multiple interconnects and multiple disk controllers. In addition, plan to write file data to separate, redundant, highly available disk devices.
Plan to write your file system's metadata to RAID-1 disks. You can write file data to either RAID-1 or RAID-5 disks.
If are configuring a Sun QFS highly available local file system and you are using a volume manager, the best performance is realized when the file system is striping data over all controllers and disks, rather than when the volume manager performs the striping. You should use a volume manager only to provide redundancy.

Example – Verifying Devices and Device Redundancy

This example shows how to use output from the `cldevice`(1CL) command to find the devices in the Solaris Cluster environment, determine which devices are highly available, and then determine which devices are redundant.

Determining High Availability

The following example shows the `cldevice show` Solaris Cluster command to list paths of the devices in the DID configuration file for all nodes. In the output from the `cldevice show` command, look for output that shows a device that is visible from two or more nodes and that has the same World Wide Name. These are global devices.

The example uses Sun StorageTek T3 arrays in a RAID-5 configuration. The output shows that you can use devices 3 and 4 for configuring the disk cache for a file system.

Example - **`cldevice`** Command Example

```
ash# cldevice show | grep Device

+=== DID Device Instances ===
DID Device Name:                                /dev/did/rdsk/d1
  Full Device Path:                               ash:/dev/rdsk/c0t0d0
DID Device Name:                                /dev/did/rdsk/d2
  Full Device Path:                               ash:/dev/rdsk/c0t6d0
DID Device Name:                                /dev/did/rdsk/d3
  Full Device Path:                               ash:/dev/rdsk/c6t50020F2300004921d1
  Full Device Path:                               elm:/dev/rdsk/c6t50020F2300004921d1
DID Device Name:                                /dev/did/rdsk/d4
  Full Device Path:                               ash:/dev/rdsk/c6t50020F2300004921d0
  Full Device Path:                               elm:/dev/rdsk/c6t50020F2300004921d0
...

     _# The preceding output indicates that both ash and elm can access DID devices {{d3}} and
{{d4}}._
     _# These disks are highly available._


ash# format /dev/did/rdsk/d4s2
selecting /dev/did/rdsk/d4s2
[disk formatted]


FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        repair     - repair a defective sector
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        inquiry    - show vendor, product and revision
        volname    - set 8-character volume name

        <cmd>     - execute <cmd>, then return
        quit
format> verify

Primary label contents:

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 192 sec 64>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =  192
nsect       =   64
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264      101.16GB    (17265/0/0) 212152320
  1        usr    wm   17265 - 34529      101.16GB    (17265/0/0) 212152320
  2     backup    wu       0 - 34529      202.32GB    (34530/0/0) 424304640
  3 unassigned    wu       0                0         (0/0/0)             0
  4 unassigned    wu       0                0         (0/0/0)             0
  5 unassigned    wu       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7 unassigned    wu       0                0         (0/0/0)             0
```

Analyzing the Output From the Commands

The `cldevice show` command in this example lists device `/dev/rdsk/c6t50020F2300004921d0`, which is DID device `/dev/did/rdsk/d4` or global device `/dev/global/rdsk/d4`. This device has two partitions (`0` and `1`), each of which yields 212152320 blocks for use by a Sun QFS highly available local file system as `/dev/global/rdsk/d4s0` and `/dev/global/rdsk/d4s1`.

You need to issue the `cldevice show` and `format` commands for all devices to be configured for use by the Sun QFS highly available local file system.

- If you want to configure a Sun QFS shared file system on a cluster, you must use highly available, redundant devices.
- If you want to configure a Sun QFS highly available local file system and the `cldevice show` command output indicates that the devices you want to use are JBOD (just a bunch of disks) or dual-port SCSI disk devices, you need to use a volume manager that is supported in a Solaris Cluster environment to obtain redundancy. The options available and capabilities provided by such a volume manager are beyond the scope of this manual.

For more information about configuring devices that are on redundant storage, see your Solaris Cluster software installation documentation.

Next Steps

- Installing Solaris Cluster Software
- Verifying Sun QFS or SAM-QFS Installation Requirements
- Installing Sun QFS File Systems
- Configuring Sun QFS File Systems With Solaris Cluster
- Installing SAM-QFS Archiving
- Configuring SAM-QFS Archiving With Solaris Cluster

Troubleshooting Issues With Oracle RAC

Contents

Troubleshooting Issue With Oracle RAC

Problem: If you use Oracle RAC with Solaris Volume Manager for shared file systems, you might see a message that SVM volumes are not available.

Solution: In response to this issue, you can configure the `rac_maxdevretry` parameter in the `/etc/opt/SUNWsamfs/defaults.conf` file. By default, the `rac_maxdevretry` variable is set to 150 (5 minutes).

The items to take in account for adjusting this variable include:

- Number of nodes in the cluster
- Number of physicial LUNs
- Numnber of SVM Diskgroups
- Speed of array topology 1G/2G/4G

In the following example `/etc/opt/SUNWsamfs/defaults.conf` file, the time is shortened to 90 (3 minutes).

```
trace
all = on
sam-sharefsd.options = all
sam-fsd.size = 10M
sam-sharefsd.size = 10M
endtrace
rac_maxdevretry = 90
```

This change was based on the following environment:

- Nodes - 8
- Physical LUNs - 32 Shared Physical LUNS w/4G Fibre Channel
- SVM disk groups - 1
- Shared file systems - 13

An example of this setting is included in `/etc/opt/SUNWsamfs/examples/defaults.conf`.

For more information about configuring an Oracle database, see the Sun Cluster Data Service for Oracle Real Application Clusters Guide for

Solaris OS.

Related Topics

- Using SAM-QFS With Solaris Cluster
- Configuring Archiving in a Solaris Cluster Environment (HA-SAM)

Complete (Printable) Guide to Using SAM-QFS With Solaris Cluster

Using SAM-QFS With Solaris Cluster

Contents

You can use Sun QFS 5.1 file systems with Solaris Cluster 3.2 1/09 software to provide failover and high availability solutions. When used with Solaris Cluster, Sun QFS file systems are limited to fewer nodes than without Solaris Cluster.

- On SPARC systems, Sun QFS with Solaris Cluster supports up to 16 nodes.
- On x64 systems, Sun QFS with Solaris Cluster supports up to 8 nodes.

Note -

- Online grow and shrink are not supported with Solaris Cluster software.
- Solaris Cluster was previously known as Sun Cluster.
- You can view all the information about working with Solaris Cluster on a single long page.

About Failover and Highly Available Configurations

You can install a Sun QFS file system in a Solaris Cluster environment and configure the file system for failover or high availability. The following configuration methods are available, depending on whether your file system is shared or unshared:

- Unshared file system - A standalone, local file system configured in a Solaris Cluster environment is considered to be a highly available local file system. When the node hosting the file system fails, the Solaris Cluster software moves the file system to another node. The highly available QFS file system uses the `SUNW.HAStoragePlus` resource type, which is distributed with the Solaris Cluster product. From the Solaris Cluster perspective, this use is referred to as a failover file system.

- Shared file system - In this configuration, one of the Solaris Cluster nodes acts as the metadata server for the file system. If the metadata server fails, the Solaris Cluster software automatically moves file system operations from the failing metadata server to a different metadata server within the cluster. Because the file system data is already shared, no additional movement is required. The shared file system configuration uses the `SUNW.qfs` resource type, which is distributed with the Sun QFS product. From the Solaris Cluster perspective, this use is simply known as a clustered file system.

Note -

- The Oracle Real Application Clusters (RAC) application is the only scalable application that the Sun QFS software supports in Solaris Cluster environments. To configure an Oracle database with Sun QFS and Solaris Cluster software, see the Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS.
- You can also have clients that are part of the shared file system but that are not part of the Solaris Cluster configuration. For more information, see Configuring Clients Outside of the Cluster.

- Shared archiving - You can also configure SAM-QFS archiving features for high availability with Solaris Cluster software (HA-SAM). When an unrecoverable problem occurs in an HA-SAM configuration, Solaris Cluster software moves the archiving and staging operations to a healthy node. HA-SAM depends on the Sun QFS Solaris Cluster agent, so this configuration must be installed with a shared file system that is mounted and managed by the Sun QFS Solaris Cluster agent. For more information, see Configuring SAM-QFS in a Sun Cluster Environment (HA-SAM).

Note -
Although installing the file system in a Solaris Cluster environment improves reliability and decreases or eliminates unplanned downtime, it does not eliminate planned downtime. To maintain the health of the file system, you might still need to bring down the Sun QFS or SAM-QFS software occasionally to run the `samfsck` process. You will also need to shut down the software to apply software patches or updates.

Task Map: Working With Solaris Cluster Software

Enabling Solaris Cluster software to work with Sun QFS or SAM-QFS involves the following tasks:

Step	Task	Description
1	Confirm Solaris Cluster-specific requirements.	The basic requirements for Sun QFS or SAM-QFS with Solaris Cluster software are very similar to those for the products without Solaris Cluster software. However, there are some specific requirements that will simplify later steps in the process. For information, see Requirements for Using SAM-QFS With Solaris Cluster.
2	Install Solaris Cluster software and configure nodes as appropriate.	If not already done, install and configure Solaris Cluster software as explained in the Solaris Cluster information products.

3	Verify Sun QFS or SAM-QFS installation requirements.	Additional requirements for installing Sun QFS or SAM-QFS software with Solaris Cluster software are basically the same as for Sun QFS and SAM-QFS software without Solaris Cluster software. See Preparing for Installation.
4	Install Sun QFS or SAM-QFS software.	See Installing and Configuring SAM-QFS or Installing Sun QFS, as applicable.
5	(Optional) Configure failover on the metadata server for clustered (shared) file systems.	See Configuring Sun QFS Shared File Systems With Solaris Cluster.
6	(Optional) Configure high availability for archiving file systems (HA-SAM).	See Configuring SAM-QFS Archiving in a Solaris Cluster Environment (HA-SAM).
7	(Optional) Configure failover for unshared, unarchiving file systems (HA-QFS).	See Configuring Sun QFS Failover File Systems With Solaris Cluster.
8	(Optional) Configure clients that are part of the shared file system but that are not part of the Solaris Cluster configuration.	See Configuring Clients Outside of the Cluster.

Requirements for Using SAM-QFS With Solaris Cluster

Consider the following requirements before you attempt to configure Solaris Cluster with Sun QFS or SAM-QFS.

Basic Product Familiarity

You must be familiar with basic Solaris Cluster concepts and operations.

In a Solaris Cluster environment, the disk space used by the file system must be configured on storage that is highly available and redundant. Ensure that you have a good understanding of the concepts in the Sun Cluster System Administration Guide for Solaris OS.
You should also be familiar with Solaris Cluster operations. For information on Solaris Cluster operations, see the following manuals:

- Sun Cluster Concepts Guide for Solaris OS
- Sun Cluster Software Installation Guide for Solaris OS
- Sun Cluster Data Services Planning and Administration Guide for Solaris OS

Hardware Requirements

You must have at least two UltraSPARC or AMD x64 hosts to use as a cluster.

> Note -
> You cannot mix hardware architectures in a shared Solaris Cluster environment. All of the nodes must be either SPARC or AMD x64.

Software Requirements

You must have the following minimum software levels installed on each cluster node:

- Solaris 10 5/08 OS
- Solaris Cluster 3.2 1/09

Each node must have the same Solaris Cluster software level and Solaris Cluster patch collection. You must install Sun QFS or SAM-QFS software packages on each node in the cluster that will host a Sun QFS file system.

Disk Device Requirements

For Clustered (Shared) File Systems

To configure a shared file system, use raw device identifier (DID) devices. In the `cldevice show` command output, these appear as `/dev/did/rdsk/dN` devices. The Solaris Cluster nodes that share the file system must have access to each DID device through a host bus adapter (HBA) direct connection. All devices must be accessible to the file system from all nodes in the Solaris Cluster environment that mount the shared file system. For more information about DID devices, see the `did`(7) man page.

When you specify these devices in your `mcf` file, use the `/dev/did/rdsk/dN` devices from the `cldevice show` command output. For more information about using the `cldevice`(1CL) command, see Example: Verifying Devices and Device Redundancy.

> Note -
> The Sun QFS software supports using multi-owner disk sets in Solaris Volume Manager for Solaris Cluster to obtain redundancy.

For Failover (Local) File Systems

To configure a local file system for failover, you must use highly available devices. You can use either of the following options:

- Raw devices - Use Solaris Cluster global devices. Use the output from the `cldevice show` command to determine the names of the global devices and substitute `global` for `did` when specifying the devices in the `mcf` file. Global devices are accessible from all nodes in a Solaris Cluster environment, even if these devices are not physically attached to all nodes. If all nodes that have a hardware connection to the disk lose their connections, the remaining nodes cannot access the disk. File systems created on global devices are not necessarily highly available.
- Volume manager - Use one of the following:
  - Solaris™ Volume Manager for Solaris Cluster for either shared or stand-alone Sun QFS configurations. Such devices are located in `/dev/md`.
  - Veritas Volume Manager (VxVM) for stand-alone QFS configurations only. Such devices are located in `/dev/vx`.

Use the `clsetup`(1CL) utility to register volume-managed devices with the Solaris Cluster framework before you configure your file system.

> Note -
> If you use a volume manager, use it only to provide redundancy. For performance reasons, do not use the volume manager to concatenate storage on separate devices; this causes the Sun QFS highly available file system to distribute I/O inefficiently across the component devices.

To confirm which devices in your Solaris Cluster environment are highly available, use the `cldevice show | grep Device` command . This command lists the paths of the devices in the DID configuration file. In the output from the `cldevice show` command, look for devices that have two or more DID devices listed with the identical DID device number. Such devices are highly available in a Solaris Cluster environment and can also be configured as global devices for a file system, even if they directly connect only to a single node.

I/O requests issued to global devices from a node other than the direct-attached node are issued over the Solaris Cluster interconnect. These single-node, global devices cease to be available when all nodes that have direct access to the device are unavailable.

Disk Device Redundancy

To implement redundancy, you have the following options:

- If you are configuring a shared file system, you can obtain redundancy through multi-owner disk sets in Solaris Volume Manager for the Solaris Cluster environment.
- If you are configuring a highly available file system, you can use either the Solaris Volume Manager or the Veritas Volume Manager to obtain mirroring (RAID-1) or striping (RAID-5).
  For more information about volume sizing and redundancy configurations, see the Solaris Volume Manager Administration Guide or your Veritas Volume Manager documentation.

Storage Redundancy

Storage redundancy is achieved by maintenance of extra disk copies of data using mirroring or RAID-1, or parity across several disks using RAID-5 to allow reconstruction of data after a disk failure. When supported by the hardware, these disk configurations enable you to configure the raw devices in a Solaris Cluster environment without a volume manager. These raw devices are accessible from multiple nodes, so you can issue the `format`(1M) command from any node to obtain information on the disks.
Storage redundancy can also be achieved by using software to support mirroring or RAID. This method, however, is not generally suitable for concurrent access from multiple hosts. Solaris Cluster software supports mirroring of disk volumes (RAID-1 only) through its multi-owner diskset feature with Sun QFS software and Solaris Volume Manager. No other software redundancy is supported.

Data Path Redundancy

Data path redundancy is achieved with multiple HBAs, which are configured from a single node. If your environment includes multiple HBAs for redundancy, be aware that the Sun QFS file systems require multipathing software like the Solaris I/O multipathing feature (MPxIO) to enable data path redundancy. For more information, see the Solaris SAN Configuration and Multipathing Guide, or see the `stmsboot`(1M) man page.

To determine redundancy, consult the hardware documentation for your disk controllers and disk devices. You need to know whether the disk controller or disk devices that are reported by the `cldevice show` command are on redundant storage. For information, see the storage controller vendor's documentation set and view the current controller configuration.

Performance Considerations

For optimal file system performance, the metadata and file data should be accessible through multiple interconnects and multiple disk controllers. In addition, plan to write file data to separate, redundant, highly available disk devices.
Plan to write your file system's metadata to RAID-1 disks. You can write file data to either RAID-1 or RAID-5 disks.
If are configuring a Sun QFS highly available local file system and you are using a volume manager, the best performance is realized when the file system is striping data over all controllers and disks, rather than when the volume manager performs the striping. You should use a volume manager only to provide redundancy.

Example – Verifying Devices and Device Redundancy

This example shows how to use output from the `cldevice`(1CL) command to find the devices in the Solaris Cluster environment, determine which devices are highly available, and then determine which devices are redundant.

Determining High Availability

The following example shows the `cldevice show` Solaris Cluster command to list paths of the devices in the DID configuration file for all nodes. In the output from the `cldevice show` command, look for output that shows a device that is visible from two or more nodes and that has the same World Wide Name. These are global devices.

The example uses Sun StorageTek T3 arrays in a RAID-5 configuration. The output shows that you can use devices 3 and 4 for configuring the disk cache for a file system.

Example - **`cldevice`** Command Example

```
ash# cldevice show | grep Device

+=== DID Device Instances ===
DID Device Name:                                /dev/did/rdsk/d1
  Full Device Path:                               ash:/dev/rdsk/c0t0d0
DID Device Name:                                /dev/did/rdsk/d2
  Full Device Path:                               ash:/dev/rdsk/c0t6d0
DID Device Name:                                /dev/did/rdsk/d3
  Full Device Path:                               ash:/dev/rdsk/c6t50020F2300004921d1
  Full Device Path:                               elm:/dev/rdsk/c6t50020F2300004921d1
DID Device Name:                                /dev/did/rdsk/d4
  Full Device Path:                               ash:/dev/rdsk/c6t50020F2300004921d0
  Full Device Path:                               elm:/dev/rdsk/c6t50020F2300004921d0
...

     _# The preceding output indicates that both ash and elm can access DID devices {{d3}} and
{{d4}}._
     _# These disks are highly available._


ash# format /dev/did/rdsk/d4s2
selecting /dev/did/rdsk/d4s2
[disk formatted]


FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        repair     - repair a defective sector
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        inquiry    - show vendor, product and revision
        volname    - set 8-character volume name

        <cmd>     - execute <cmd>, then return
        quit
format> verify

Primary label contents:

Volume name = <        >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 192 sec 64>
pcyl        = 34532
ncyl        = 34530
acyl        =    2
nhead       =  192
nsect       =   64
Part      Tag    Flag     Cylinders         Size            Blocks
  0        usr    wm       0 - 17264      101.16GB    (17265/0/0) 212152320
  1        usr    wm   17265 - 34529      101.16GB    (17265/0/0) 212152320
  2     backup    wu       0 - 34529      202.32GB    (34530/0/0) 424304640
  3 unassigned    wu       0                0         (0/0/0)             0
  4 unassigned    wu       0                0         (0/0/0)             0
  5 unassigned    wu       0                0         (0/0/0)             0
  6 unassigned    wu       0                0         (0/0/0)             0
  7 unassigned    wu       0                0         (0/0/0)             0
```

Analyzing the Output From the Commands

The `cldevice show` command in this example lists device `/dev/rdsk/c6t50020F2300004921d0`, which is DID device `/dev/did/rdsk/d4` or global device `/dev/global/rdsk/d4`. This device has two partitions (`0` and `1`), each of which yields 212152320 blocks for use by a Sun QFS highly available local file system as `/dev/global/rdsk/d4s0` and `/dev/global/rdsk/d4s1`.

You need to issue the `cldevice show` and `format` commands for all devices to be configured for use by the Sun QFS highly available local file system.

- If you want to configure a Sun QFS shared file system on a cluster, you must use highly available, redundant devices.
- If you want to configure a Sun QFS highly available local file system and the `cldevice show` command output indicates that the devices you want to use are JBOD (just a bunch of disks) or dual-port SCSI disk devices, you need to use a volume manager that is supported in a Solaris Cluster environment to obtain redundancy. The options available and capabilities provided by such a volume manager are beyond the scope of this manual.

For more information about configuring devices that are on redundant storage, see your Solaris Cluster software installation documentation.

Configuring Sun QFS Local Failover File Systems With Solaris Cluster

This section contains instructions for configuring the Sun QFS software in a Solaris Cluster environment. Before carrying out the configuration procedures in this section, you must have installed the Sun QFS software as described in Installing Sun QFS.

> Note -
> The Oracle Real Application Clusters (RAC) application is the only scalable application that the Sun QFS software supports in Solaris Cluster environments. For more information, see the Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS. See also Troubleshooting Issues With Oracle RAC.

The Sun QFS file system can work with Solaris Cluster in one of the following ways:

- As a highly-available shared file system, simply referred to as a clustered file system. For information about configuring a clustered file system, see Configuring Sun QFS Shared File Systems With Solaris Cluster
- As a highly-available local file system, also known as a failover file system.

Task Map: Configuring Local Failover File Systems With Solaris Cluster

Step	Task	Description
1	Prepare the host systems.	Verify user and group IDs, and NFS-share the file system as described in Preparing the Host Systems.
2	Edit the `mcf` file.	Edit the `mcf` file for each host that you want to include in the Solaris Cluster environment. See Editing mcf Files for a Local Failover File System.
3	Make the highly-available file system available to Solaris Cluster.	Configure the resource as described in How to Configure a Highly Available File System as a SUNW.HAStoragePlus Resource.
4	Verify the resource group on all nodes.	From any node in the cluster, move the highly-available resource to another node as described in How to Verify the Resource Group on All Nodes.

Preparing the Host Systems

To prepare the host systems for a shared file system in a Solaris Cluster environment, perform the following tasks:

1. Verify that all the hosts have the same user and group IDs.
    - If you are not running the Network Information Name service (NIS), make sure that all `/etc/passwd` and all `/etc/group` files are identical.
    - If you are running NIS, the `/etc/passwd` and `/etc/group` files should already be identical.
      For more information, see the `nis+`(1) man page.
2. NFS-share the file system.
   The following is a general description of how to NFS-share a file system in a Solaris Cluster environment. For more information on NFS-sharing file systems that are controlled by HAStoragePlus, see the Sun Cluster Data Service for Network File System (NFS) Guide for Solaris OS, and your NFS documentation.
    a. Locate the `dfstab.`resource-name file.
       The `Pathprefix` property of HAStoragePlus specifies the directory in which the `dfstab.`resource-name file resides.
    b. Use `vi`(1) or another editor to add a `share`(1M) command to the Pathprefix`/SUNW.nfs/dfstab.`resource-name file.
       For example:

```
share -F nfs -o rw /global/qfs1
```

Editing **mcf** Files for a Local Failover File System

The lines that define a particular file system must be identical in the `mcf` files on all host systems that support the file system. Only one `mcf` file can reside on a host. Because you can have other, additional file systems defined in an `mcf` file, the `mcf` files on different hosts might not be identical.

> Note -
> If you update a metadata server's `mcf` file at any time after the shared file system is mounted, you must also update the `mcf` files as necessary on all hosts that can access that shared file system.

How to Edit mcf Files for a Failover File System

Perform this procedure to enable a local file system to failover in a Solaris Cluster environment. Perform these steps on each host that supports the file system.

1. Log in to the Solaris Cluster node.
2. Become superuser.
3. Use `vi`(1) or another editor to create an `mcf` file on that node.
   If an `mcf` file already exists on the host, add the lines for the new file system to this `mcf` file.
4. Copy the lines that define the file system from the primary node's `mcf` file to this node's `mcf` file.

Examples

Example 1 – Solaris OS Hosts

This example shows how the `samfsconfig`(1M) command is used to retrieve device information for family set `sharefs1` on client `tethys`. Because `tethys` is a potential metadata server, it is connected to the same metadata disks as `titan`.

Example – **samfsconfig**(1M) Command Example on **tethys**

```
tethys# samfsconfig /dev/dsk/*
#
# Family Set "sharefs1" Created Wed Jun 27 19:33:50 2003
#
sharefs1 10 ma sharefs1 on shared
/dev/dsk/c2t50020F23000065EEd0s6 11 mm sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12 mr sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13 mr sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14 mr sharefs1 on
```

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

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

Example – **mcf** File for **sharefs1** Client Host **tethys**

```
# Equipment Eq Eq Family Dev Add
# Identifier Ord Type Set State Params
# ---------- --- ---- ------ ----- ------
sharefs1 10 ma sharefs1 on shared
/dev/dsk/c2t50020F23000065EEd0s6 11 mm sharefs1 on
/dev/dsk/c7t50020F2300005D22d0s6 12 mr sharefs1 on
/dev/dsk/c7t50020F2300006099d0s6 13 mr sharefs1 on
/dev/dsk/c7t50020F230000651Cd0s6 14 mr sharefs1 on
```

The Equipment Number numbers match those of the `mcf` file for metadata server `titan` (see Configuration Example for a Shared File System on a Solaris OS Platform). These Equipment Number numbers must not already be in use on client host `tethys` or any other client host.

Example 2 – Solaris OS Hosts

This example shows how the `samfsconfig`(1M) command is used to retrieve device information for family set `sharefs1` on client host `mimas`. Because `mimas` can never become a metadata server, it is not connected to the metadata disks.

Example – **`samfsconfig`**(1M) Command Example on **`mimas`**

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

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

- Create a file system Family Set declaration line, beginning with `sharefs1`, in the `mcf` file. Enter the `shared` keyword in the Additional Parameters field of the file system Family Set declaration line.
- Create one or more `nodev` lines for each missing Equipment Number field. For these lines, the keyword `nodev` must appear in the Equipment Identifier field for each inaccessible device. In this example, you create a device entry in the `mcf` file named `nodev` to represent the missing metadata disk.
- Ensure that each Device State field is set to on.
- Uncomment the device lines.
  The following example shows the resulting `mcf` file.

Example – **`mcf`** File for Client Host **`mimas`**

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

How to Configure a Failover File System as a SUNW.HAStoragePlus Resource

Perform this task if you are configuring a local file system to failover on a Solaris Cluster platform.

1. Use the `scrgadm`(1M) command to set the `FilesystemCheckCommand` property of HAStoragePlus to `/bin/true`.
   All other resource properties for HAStoragePlus apply as specified in `SUNW.HAStoragePlus`(5).
   The following example command shows how to use the `scrgadm`(1M) command to configure a `SUNW.HAStoragePlus` resource:

```
# scrgadm -a -g qfs-rg -j ha-qfs -t SUNW.HAStoragePlus \
        -x FilesystemMountPoints=/global/qfs1 \
        -x FilesystemCheckCommand=/bin/true
```

How to Verify the Resource Group on All Nodes

This task ensures that the file system can move from node to node when the Solaris Cluster software performs a failover.

Perform these steps for each node in the cluster, with a final return to the original server.

Steps

1. From any node in the Solaris Cluster environment, use the `scswitch`(1M) command to move the file system resource from one node to another.
   For example:

```
server# scswitch -z -g qfs-rg -h elm
```

2. Use the `scstat`(1M) command to verify that the file system resource was moved successfully.
   For example:

```
server# scstat
-- Resources --
Resource Name    Node Name  State     Status Message
-------------    ---------  -----     --------------
Resource: qfs-res   ash     Offline   Offline
Resource: qfs-res   elm     Online    Online
Resource: qfs-res   oak     Offline   Offline
```

Configuring Sun QFS Shared File Systems With Solaris Cluster

This section contains instructions for configuring the Sun QFS software in a Solaris Cluster environment. Before carrying out the configuration procedures in this section, you must have installed the Sun QFS software as described in Installing Sun QFS.

Note -

- You can also have clients that are part of the shared file system but that are not part of the Solaris Cluster configuration. For more information, see Configuring Clients Outside of the Cluster.
- The Oracle Real Application Clusters (RAC) application is the only scalable application that the Sun QFS software supports in Solaris Cluster environments. For more information, see the Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS. See also Troubleshooting Issues With Oracle RAC.

The Sun QFS file system can work with Solaris Cluster in one of the following ways:

- As a highly-available shared file system, simply referred to as a clustered file system.
- As a highly-available local file system, also known as a failover file system. For information about configuring a failover file system, see Configuring Sun QFS Failover File Systems With Solaris Cluster.

You can also add a shared file system to an existing Solaris Cluster environment. For information, see Adding and Removing Nodes.

Task Map: Configuring Clustered File Systems With Solaris Cluster

Step	Task	Description

1	Prepare the host systems.	Verify user and group IDs, and NFS-share the file system as described in Preparing the Host Systems.
2	Edit the `mcf` file.	Edit the `mcf` file for each host that you want to include in a shared file system in a Solaris Cluster environment. See Editing mcf Files for a Clustered File System.
3	Create the shared hosts file.	On the metadata server, create the shared hosts file as described in Creating the Shared Hosts File on the Metadata Server.
4	Verify that the shared file system is mounted.	On each host that can mount the file system, verify that the file system is mounted and shared. See How to Verify the Daemons.
5	Make the shared file system available to Solaris Cluster.	Log in to the metadata server and configure the resource as described in How to Enable a Shared File System as a SUNW.qfs(5) Resource.
6	Bring the shared resource online.	On the metadata server, move the shared resource to another node as described in How to Bring the Shared Resource Online.
7	Verify the resource group on all nodes.	On each node in the cluster that is part of the shared file system, move the shared resource to another node as described in How to Verify the Resource Group on All Nodes. When you are finished, return to the metadata server.

Preparing the Host Systems

To prepare the host systems for a shared file system in a Solaris Cluster environment, perform the following tasks:

1. Verify that all the hosts have the same user and group IDs.
    - If you are not running the Network Information Name service (NIS), make sure that all `/etc/passwd` and all `/etc/group` files are identical.
    - If you are running NIS, the `/etc/passwd` and `/etc/group` files should already be identical.
      For more information, see the `nis+`(1) man page.
2. NFS-share the file system.
   The following is a general description of how to NFS-share a file system in a Solaris Cluster environment. For more information on NFS-sharing file systems that are controlled by HAStoragePlus, see the Sun Cluster Data Service for Network File System (NFS) Guide for Solaris OS, and your NFS documentation.
    a. Locate the `dfstab.`resource-name file.
       The `Pathprefix` property of HAStoragePlus specifies the directory in which the `dfstab.`resource-name file resides.
    b. Use `vi`(1) or another editor to add a `share`(1M) command to the Pathprefix`/SUNW.nfs/dfstab.`resource-name file.
       For example:

```
share -F nfs -o rw /global/qfs1
```

Editing **`mcf`** Files for a Clustered File System

The lines that define a particular file system must be identical in the `mcf` files on all host systems that support the file system. Only one `mcf` file can reside on a host. Because you can have other, additional file systems defined in an `mcf` file, the `mcf` files on different hosts might not be identical.

> Note -
> If you update a metadata server's `mcf` file at any time after the shared file system is mounted, you must also update the `mcf` files as necessary on all hosts that can access that shared file system.

How to Edit mcf Files for a Clustered File System

Perform this procedure for each host that you want to include in a shared file system in a Solaris Cluster environment.

1. Log in to the host.

2. Become superuser.
3. Use the `format`(1M) command to verify the presence of client host disks.
4. Use `vi`(1) or another editor to create an `mcf` file.
   If an `mcf` file already exists on the host, add the lines for the new file system to this `mcf` file.
5. Use the `samfsconfig`(1M) command to locate the local device names for each additional host to be configured in the shared file system.
   The `samfsconfig`(1M) command generates configuration information that can help you to identify the devices included in the shared file system. Enter a separate `samfsconfig`(1M) command on each client host. Note that the controller number might not be the same controller number as on the metadata server because the controller numbers are assigned by each client host.
6. Update the `mcf` file on other client hosts.
   To access or mount a shared file system, a host system must have that file system defined in its `mcf` file. The content of the `mcf` files varies, depending on the type of Solaris Cluster hosts:
   - The primary metadata server
   - Potential metadata servers
     All hosts can be metadata servers because the Solaris Cluster software fails over system resources in the event of a node failure.
     Use `vi`(1) or another editor to edit the `mcf` file on one of the client host systems. The `mcf` file must be updated on all client hosts to be included in the shared file system. The file system and disk declaration information must have the same data for the Family Set name, Equipment Number, and Equipment Type fields as the configuration on the metadata server. The `mcf` files on the client hosts must also include the `shared` keyword. The device names, however, can change as controller assignments can change from host to host.

Example 1 – **`mcf`** File Using Hardware RAID and Solaris Cluster

The following sample `mcf` file includes four shared file systems. The first three are on Solaris Cluster nodes. The last file system is not part of the cluster.

```
psfb1% more /etc/opt/SUNWsamfs/mcf
OraHomes 20 ma OraHomes - shared
/dev/did/dsk/d1s0 21 mm OraHomes -
/dev/did/dsk/d2s0 22 mr OraHomes -
/dev/did/dsk/d3s0 23 mr OraHomes -

CrsHomes 30 ma CrsHomes - shared
/dev/did/dsk/d6s0 31 mm CrsHomes -
/dev/did/dsk/d7s0 32 mr CrsHomes -

OraData 40 ma OraData - shared
/dev/did/dsk/d8s0 41 mm OraData -
/dev/did/dsk/d9s0 42 mr OraData -
/dev/did/dsk/d10s0 43 mr OraData -
/dev/did/dsk/d11s0 44 mr OraData -

votingdisk1 50 ms votingdisk1 - shared
/dev/did/dsk/d4s1 51 md votingdisk1 -
votingdisk2 60 ms votingdisk2 - shared
/dev/did/dsk/d4s2 61 md votingdisk2 -
votingdisk3 70 ms votingdisk3 - shared
/dev/did/dsk/d4s3 71 md votingdisk3 -
```

Creating the Shared Hosts File on the Metadata Server

The shared hosts configuration file must reside in the following location on the metadata server:

```
/etc/opt/SUNWsamfs/hosts.<fsname>
```

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

The following table shows the fields in the hosts configuration file.

Table – Shared Hosts Configuration File Fields

Field	Content
Host Name	This field must contain the alphanumeric name of a metadata server or potential metadata server that is part of the Sun QFS shared file system.
Host Interfaces	This field must contain a comma-separated list of host interface addresses. This field can be created from the output received from the `ifconfig`(1M) `-a` command. The individual interfaces can be specified in one of the following ways: * Dotted-decimal IP address form * IP version 6 hexadecimal address form * As a symbolic name that the local domain name service (DNS) can resolve to a particular host interface Each host uses this field to determine whether it will try to connect to the specified host interface. The system evaluates the addresses from left to right, and the connection is made using the first responding address in the list that is also included in the shared hosts file.

In a shared file system, each client host obtains the list of metadata server IP addresses from the metadata server host.

The metadata server and the client hosts use both the `/etc/opt/SUNWsamfs/hosts.`fsname file on the metadata server and the `hosts.`fsname`.local` file on each client host (if it exists) to determine the host interface to use when accessing the file system.

How Metadata Server Addresses Are Obtained

The information in this section might be useful when you are debugging.

In a shared file system, each client host obtains the list of metadata server IP addresses from the shared hosts file.
The metadata server and the client hosts use the shared hosts file on the metadata server and the `hosts.`fsname`.local` file on each client host (if it exists) to determine the host interface to use when accessing the metadata server. This process is as follows:

> Note -
> The term client, as in network client, is used to refer to both client hosts and the metadata server host.

1. The client obtains the list of metadata server host IP interfaces from the file system's on-disk shared hosts file. To examine this file, issue the `samsharefs`(1M) command from the metadata server or from a potential metadata server.
2. The client searches for an `/etc/opt/SUNWsamfs/hosts.`fsname`.local` file. Depending on the outcome of the search, one of the following occurs:
    - If a `hosts.`fsname`.local` file does not exist, the client attempts to connect, in turn, to each address in the server's line in the shared hosts file until it succeeds in connecting.
    - If the `hosts.`fsname`.local` file exists, the client performs the following tasks:
3. It compares the list of addresses for the metadata server from both the shared hosts file on the file system and the `hosts.`fsname `.local` file.
4. It builds a list of addresses that are present in both places, and then it attempts to connect to each of these addresses, in turn, until it succeeds in connecting to the server. If the order of the addresses differs in these files, the client uses the ordering in the `hosts.`fsname`.local` file.

Example – Sun QFS Shared File System Hosts File Example

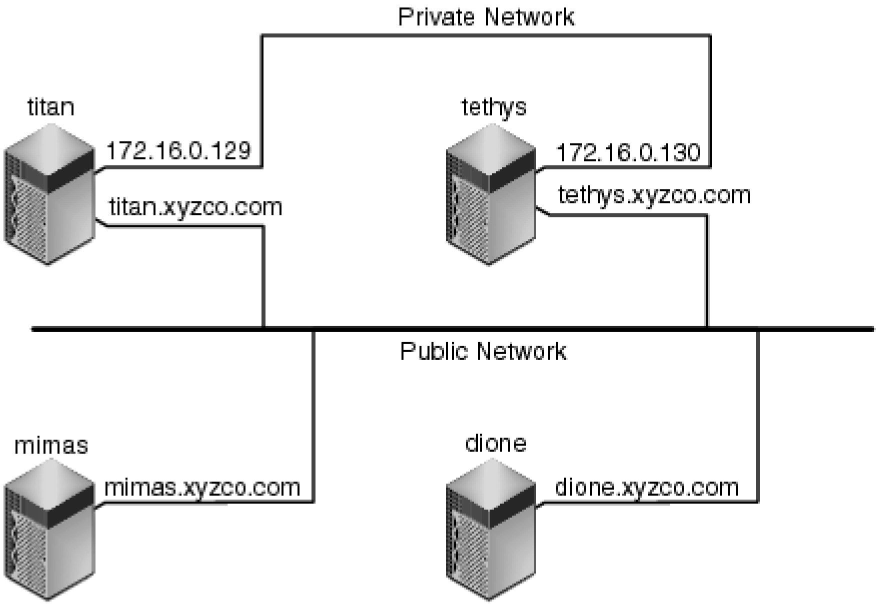
This set of examples shows a detailed scenario for a shared file system that comprises four hosts.

The following example shows a hosts file that lists four hosts.

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IP Server Not Server
# Name Addresses Priority Used Host
# ---- ---------------- -------- ---- -----
titan 172.16.0.129 1 - server
tethys 172.16.0.130 2 -
mimas mimas - -
dione dione - -
```

The following figure shows the interfaces to these systems.

Figure – Network Interfaces for Sun QFS Shared File System Hosts File Example



Example – File **`hosts.sharefs1.local`** on **`titan`** and **`tethys`**

Systems `titan` and `tethys` share a private network connection with interfaces `172.16.0.129` and `172.16.0.130`. To guarantee that `titan` and `tethys` always communicate over their private network connection, the system administrator has created identical copies of `/etc/opt/SUNWsamfs/hosts.sharefs1.local` on each system.

The following example shows the information in the `hosts.sharefs1.local` files on `titan` and `tethys`.

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

Example – File **`hosts.sharefs1.local`** on **`mimas`** and **`dione`**

Systems `mimas` and `dione` are not on the private network. To guarantee that they always connect to `titan` and `tethys` through `titan`'s and `tethys`'s public interfaces, the system administrator has created identical copies of `/etc/opt/SUNWsamfs/hosts.sharefs1.local` on `mimas` and `dione`.

The following example shows the information in the `hosts.sharefs1.local` files on `mimas` and `dione`.

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

How to Verify the Daemons

Perform this task on each host that can mount the file system.

1. Verify that the file system is mounted.
   If it is not mounted, go back to Mounting the File System and follow the instructions there.
2. Use the `ps`(1) and `grep`(1) commands to determine whether the `sam-sharefsd` daemon is running for this file system.
   For example:

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

This example shows that the `sam-sharefsd` daemon is active for the `sharefs1` file system.

> Note -
> If the `sam-sharefsd` daemon is active for your shared file system, you need to perform some diagnostic procedures. For information about these procedures, see Using Daemons, Processes, and Tracing.

3. If the output from this command indicates that the `sam-sharefsd` daemon is not running, determine whether the `sam-fsd` daemon is running as follows:
   a. Use the `ps`(1) and `grep`(1) commands to verify that the `sam-fsd` daemon is running for this file system.
   b. Examine the output.
      The following code example shows `sam-fsd` output that indicates that the daemon is running.

      ```
      cur% ps -ef | grep sam-fsd
      user1 16435 16314  0 16:52:36 pts/13   0:00 grep sam-fsd
      root   679     1  0   Aug 24 ?        0:00 /usr/lib/fs/samfs/sam-fsd
      ```

   c. Do one of the following:
      - If the output indicates that the `sam-fsd` daemon is not running, and if no file system has been accessed since the system's last boot, issue the `samd`(1M) `config` command, as follows:

        ```
        # samd config
        ```

      - If the output indicates that the `sam-fsd` daemon is running, enable tracing in the `defaults.conf` file and check the following files to determine whether configuration errors are causing the problem:
        - `/var/opt/SUNWsamfs/trace/sam-fsd`
        - `/var/opt/SUNWsamfs/trace/sam-sharefsd`

How to Enable a Shared File System as a SUNW.qfs(5) Resource

1. Log in to the metadata server as superuser.
2. Use the `scrgadm`(1M) `-p` command to search for the `SUNW.qfs`(5) resource type.
   For example:

   ```
   metadataserver# scrgadm -p | grep SUNW.qfs
   ```

3. If the `SUNW.qfs` resource type is missing, issue the following command:

   ```
   metadataserver# scrgadm -a -t SUNW.qfs
   ```

4. Use the `scrgadm`(1M) command to set the `FilesystemCheckCommand` property of the `SUNW.qfs`(5) resource type to `/bin/true`
   .
   The `SUNW.qfs`(5) resource type is part of the Sun QFS software package. Configuring the resource type for use with your shared file system makes the shared file system's metadata server highly available. Solaris Cluster scalable applications can then access data contained in the file system.
   The following code example shows how to use the `scrgadm`(1M) command to register and configure the `SUNW.qfs` resource type. In this example, the nodes are `scnode-A` and `scnode-B`. `/global/sharefs1` is the mount point as specified in the `/etc/vfstab` file.

```
# scrgadm -a -g qfs-rg -h scnode-A,scnode-B
# scrgadm -a -g qfs-rg -t SUNW.qfs -j qfs-res \
       -x QFSFileSystem=/global/sharefs1
```

Note -
In a SAM-QFS environment, you can also configure the archiving features for high availability using Solaris Cluster software. For instructions, see Configuring Archiving in a Solaris Cluster Environment (HA-SAM).

How to Bring the Shared Resource Online

1. Verify that the file system is mounted on all nodes.

   Note -
   If you are using the `SUNW.qfs` resource type, you cannot use the bg mount option in the `/etc/vfstab` file.

2. Log in to the node upon which the file system is based.
3. Use the `scswitch`(1M) command to move the file system resource to another node.
   For example:

```
metadataserver# scswitch -Z -g qfs-rg
```

4. Use the `scstat`(1M) command to verify that the file system resource was moved successfully.
   For example:

```
metadataserver# scstat

<information deleted from this output>

-- Resources --
Resource Name     Node Name  State      Status Message
-------------     ---------  -----      --------------
Resource: qfs-res   ash      Online     Online
Resource: qfs-res   elm      Offline    Offline
Resource: qfs-res   oak      Offline    Offline
```

How to Verify the Resource Group on All Nodes

Perform these steps for each node in the cluster, with a final return to the metadata server.

Steps

1. From any node in the Solaris Cluster environment, use the `scswitch`(1M) command to move the file system resource from one node to another.
   For example:

```
server# scswitch -z -g qfs-rg -h elm
```

2. Use the `scstat`(1M) command to verify that the file system resource was moved successfully.
   For example:

```
server# scstat
-- Resources --
Resource Name    Node Name  State     Status Message
-------------    ---------  -----     --------------
Resource: qfs-res   ash     Offline   Offline
Resource: qfs-res   elm     Online    Online
Resource: qfs-res   oak     Offline   Offline
```

Configuring SAM-QFS Archiving in a Solaris Cluster Environment (HA-SAM)

About HA-SAM

High availability Sun Storage Archive Manager (HA-SAM) is an interface between a Sun QFS file system and Solaris Cluster software running on Solaris for SPARC and x64 hardware. The HA-SAM Solaris Cluster agent periodically monitors the health of SAM-QFS archiving operations on the primary node. In the event of an unrecoverable problem, the agent switches the SAM-QFS archiving and staging operations to a healthy node. Both voluntary and involuntary failover are supported on active-passive configurations. Only two-node active-passive configurations are supported.

For tape archiving and staging to continue after failover, tape drives must be visible to all nodes in a cluster on which HA-SAM is running, but they should not be configured as SAM-QFS shared drives. HA-SAM also supports disk archiving with disk archives visible to all nodes in a cluster.

HA-SAM depends on the Sun QFS Solaris Cluster agent and assumes that the Sun QFS file systems are mounted and managed by the Sun QFS agent. HA-SAM requires that the SAM-QFS catalog and stager directories be linked from the standard location to a directory in an HAStoragePlus file system. HA-SAM should be a resource in a resource group that contains Sun QFS and catalog resources.

In order to configure the HA-SAM Solaris Cluster agent, the Sun QFS Solaris Cluster agent must already be configured. These instructions assume that you have QFS configured in a Solaris Cluster environment using a shared QFS file system as described in Configuring Sun QFS Shared File Systems With Solaris Cluster.

Before You Begin

The following are requirements and restrictions that you should be aware of before configuring this feature.

- The nodes on which HA-SAM will be configured must be running the same version of Solaris (10) and Solaris Cluster (3.2 U2 or newer). Mixed versions of Solaris or Solaris Cluster software are not supported. No operating system other than Solaris is supported.
- The nodes in the cluster running HA-SAM must have the same type of architecture: SPARC or x64. Mixed architectures are not supported.
- No more than two nodes can be configured.
- Only active-passive configurations are supported, not active-active.
- Active I/O to HA-SAM file systems is supported only on the active node of an HA-SAM file system.
- Only shared QFS file systems are supported. Both `ma`- and `ms`-type file systems are supported. Stand-alone QFS environments are not supported.
- No software volume managers are supported with this configuration.
- Within the HA-SAM environment, the `nosam` mount option must be specified for any non-HA-SAM QFS file systems.
- The HA-SAM resource, QFS file systems, and HAStoragePlus file systems must all be configured within the same resource group. A separate Solaris Cluster resource group must be created for non-HA-SAM file systems.
- When using the `SUNW.hasam` resource type, you cannot specify the `bg` mount option in the `/etc/vfstab` file.
- Fibre tape drives are required. Tape drives must be visible to all systems through the fibre fabric, but should not be configured as SAM-QFS shared drives.
- Disk volumes for disk archiving must be visible to all nodes.
- The active metadata server and potential metadata server must not be configured as a SAM-Remote client or server.
- The SAM-QFS catalog and stager directory must be in the default location: `/var/opt/SUNWsamfs/`. If it is any other location, the cluster nodes will not be able to locate it.
- Before configuring HA-SAM, verify that all SAM-QFS archiving operations are working correctly on the required nodes in the cluster.
- Only highly available (HA) agents are supported; no scalable agents are supported.
- Oracle software is not supported with this configuration, but the HA-SAM agent is supported as an RMAN target for Oracle backups.
- Voluntary failover forces a failover after five minutes in order to avoid problems with other potential HA agents.
- During voluntary failover, active archiving and staging times-out and terminates after less than 5 minutes.
- During failover, all SAM-QFS file systems must be failed over in order to avoid complications with the recycler that could result in loss of data.

- Sun StorageTek network attached tape libraries are supported, but not ADIC or Sony 8400 PetaSite Series automated tape libraries.
- HA-SAM environments cannot be managed by the SAM-QFS Manager browser interface.
- During involuntary failover, error messages for idled tape drives may be seen in the `/var/adm/messages` log file. These messages can safely be ignored. They simply indicate that HA-SAM did not have time to idle the drives before failover.
- After failover, volumes in the catalog may be marked with the E flag. This can occur when the software is unable to write the correct label at the end of a tape. To clear this error, see How to Clear Media Errors.

How to Confirm the Catalog and Stager Symbolic Links

In order to configure for high availability, the SAM-QFS catalog and stager queues must be in an HAStoragePlus file system. During the software installation process symbolic links should have been created from `/var/opt/SUNWsamfs` to the shared HAStoragePlus file system. This must be confirmed before continuing with the configuration process.

1. Verify that the `/var/opt/SUNWsamfs/catalog` and `/var/opt/SUNWsamfs/stager` contain a symbolic link to the HAStoragePlus mount point.
2. View the catalog and stager files.

```
# ls -l /var/opt/SUNWsamfs/catalog /var/opt/SUNWsamfs/stager
```

3. Verify that the output shows symbolic links similar to the following example.

```
lrwxrwxrwx 1 root other 19 Sep 30 11:05
/var/opt/SUNWsamfs/catalog -> /sam_shared/catalog
lrwxrwxrwx 1 root other 18 Sep 30 11:05
/var/opt/SUNWsamfs/stager -> /sam_shared/stager
```

If the symbolic links exist, proceed to How to Configure SAM-QFS Archiving in a Sun Cluster Environment. If the links do not exist, proceed to the next step.

4. Copy any existing SAM-QFS catalog information to a temporary location.
5. Delete the existing `catalog` and `stager` files from their current location.

```
# rm -rf /var/opt/SUNWsamfs/catalog /var/opt/SUNWsamfs/stager
```

6. Create the `/var/opt/SUNWsamfs/catalog` and `/var/opt/SUNWsamfs/stager` symbolic links to the HAStoragePlus mount point.
The following example is for an HAStoragePlus file system mounted at `/sam_shared`.

```
# ln -s /sam_shared/catalog /var/opt/SUNWsamfs/catalog
# ln -s /sam_shared/stager /var/opt/SUNWsamfs/stager
```

7. Copy the catalog information saved in Step c to the new shared catalog.

```
# cp -rp /var/tmp/catalog/* /var/opt/SUNWsamfs/catalog
```

How to Configure SAM-QFS Archiving in a Solaris Cluster Environment

In this example procedure, two file systems are configured: `qfs1` and `qfs2`.

1. Create an HA-SAM resource group.

```
# clresourcegroup create -n scnode-A,scnode-B SAM-HA
```

2. Create and configure the `SUNW.qfs` resource.

```
# clresource create -g SAM-HA -t SUNW.qfs \
-x QFSFileSystem=/global/qfs1,/global/qfs2 qfs-res
```

3. Create and configure a SAM-QFS shared resource.

```
# clresource create -g SAM-HA -t SUNW.HAStoragePlus \
-x FilesystemMountPoints=/sam_shared -x AffinityOn=TRUE sam-hastp
```

4. Create the `HA-SAM` resource type.

```
# clresourcetype register SUNW.hasam
```

5. Create and configure the `SUNW.hasam` resource.

```
# clresource create -g SAM-HA -t SUNW.hasam -x QFSName=qfs1,qfs2 \
-x CatalogFileSystem=/sam_shared sam-ha
```

6. Create dependencies between resources within the resource group.

```
# clresource set -p Resource_dependencies=sam-hastp qfs-ha
# clresource set -p Resource_dependencies=qfs-ha sam-ha
```

7. Bring the resource group online.

```
# clresourcegroup manage SAM-HA
# clresourcegroup online -eM SAM-HA
```

> Note
> The QFS file systems must be mounted before bringing the resource group online.

8. Ensure that the resource group is functional on all configured nodes.

```
# clresourcegroup switch -n scnode-B SAM-HA
# clresourcegroup switch -n scnode-A SAM-HA
```

Using the **`samd hastop`** Command

The `samd` command has an option that is specifically for use with HA-SAM. The `samd hastop` command stops the archiver and stager daemons before stopping `sam-amld` and its children.

The `samd hastop` command must not be used by an administrator in either a stand-alone or shared SAM-QFS environment. In addition, the normal `samd stop` command must not be used on nodes under HA-SAM control.

Using Sun StorageTek Libraries with HA-SAM

If you are using Sun StorageTek libraries in an HA-SAM environment, the STK daemons need to be notified to use a forced dismount instead of a normal dismount in the case of an involuntary failover. To accomplish this, the `/var/run/hasam_running` file is created on the active Solaris Cluster node when the HA-SAM resource is brought online. This is a zero-byte file with `root` permissions. If this file is detected on a node, `sam-stkd` and `sam-stk_helper` use `force_media` to dismount the tape on the drive. This allows the successful initialization of the library and the drives after an involuntary failover. The `hasam_running` file is deleted when a SAM-QFS resource is brought offline on a node.

In order to ensure the correct failover behavior for Sun StorageTek libraries, make sure the `/var/run/hasam_running` is not deleted if found on a Solaris Cluster node that has an HA-SAM resource online.

Configuring Clients Outside of the Cluster

You can configure a Sun QFS shared file system operating in a Solaris Cluster environment to support automated metadata server failover while providing file system access for clients that are not members of the Solaris Cluster configuration. This feature is commonly referred to as Clients Outside of The Cluster (COTC).

> Note -
> Before performing any COTC tasks, make sure that you are familiar with information about shared and local host tables.

COTC Task Map

You must complete the following procedures.

Step	Task	Description
1	Verify prerequisites and identify clients that are to be configured outside of the cluster.	See Configuration Prerequisites
2	Install Solaris Cluster software.	If not already done, install the Solaris Cluster software as explained in the Solaris Cluster information products.
3	Install Sun QFS software on the metadata server and shared clients.	See Installing Sun QFS.
4	Configure the metadata server.	See Configuring the Metadata Server.
5	Configure the clients.	See Configuring the File System Clients.

Configuration Requirements

The following table identifies specific configuration rules for each segment of the COTC environment:

Sun QFS Metadata Server	• `mm` devices must be on a separate physical LUN from data devices • `mr` devices must be on a separate physical LUN from metadata devices • `mm` devices must not have the Solaris Cluster "localonly" flag
Sun QFS Clients	• `mm` devices must be identified as `nodev` in the `mcf` file • `mr` devices must follow the `/dev/dsk` syntax, rather than the `/dev/did/dsk` syntax • The definition for the `mr` device on the client `/dev/dsk` must match the definition for the `/dev/did/dsk` on the metadata server
Solaris Cluster Devices	• Set "localonly" flag on `mr` devices
Private Metadata Network	• Solaris Cluster nodes that serve as metadata servers should use IPMP groups • Solaris Cluster Sun QFS resource groups should have a logical host name configured for metadata communication betweeen the metadata server and the clients that are outside of the cluster • Solaris Cluster Sun QFS resource groups should be defined to ensure that Sun QFS file system resources depend on the logical host name resource

How to Configure the Metadata Server

The following procedure includes embedded examples to better explain what you should expect to see in a COTC configuration.

1. Use the Solaris Cluster command `cluster show -t device` to identify the devices that will be used for the Sun QFS configuration and build the `mcf` file.

Note the use of the `/dev/did` syntax in the following sample `mcf` file for this configuration.

```
#
# File system Qfs1
#
Qfs1 2 ma Qfs1 on shared
/dev/did/dsk/d7s0 20 mm Qfs1 on
/dev/did/dsk/d8s0 21 mm Qfs1 on
/dev/did/dsk/d16s0 22 mr Qfs1 on
/dev/did/dsk/d10s0 23 mr Qfs1 on
/dev/did/dsk/d13s0 24 mr Qfs1 on
#
# File system Qfs2
#
Qfs2 5 ma Qfs2 on shared
/dev/did/dsk/d9s0 50 mm Qfs2 on
/dev/did/dsk/d11s0 51 mm Qfs2 on
/dev/did/dsk/d17s0 52 mr Qfs2 on
/dev/did/dsk/d12s0 53 mr Qfs2 on
/dev/did/dsk/d14s0 54 mr Qfs2 on
/dev/did/dsk/d15s0 55 mr Qfs2 on
/dev/did/dsk/d18s0 56 mr Qfs2 on
```

Example **`/etc/opt/SUNWsamfs/defaults.conf`** File

```
trace
all = on
sam-fsd.size = 10M
sam-sharefsd.size = 10M
endtrace
```

Example **`/etc/opt/SUNWsamfs/samfs.cmd`** File

```
fs = Qfs1
meta_timeo=0

fs = Qfs2
meta_timeo=0
```

2. Build the Sun QFS file system hosts tables.
   The example configuration uses shared and local file system host files for this QFS shared file system setup.

   > Note -
   > To communicate with metadata clients that are outside of the cluster, you must establish Sun QFS metadata traffic over the Sun QFS network. Because the metadata client is not a member of the Solaris Cluster configuration, the example uses a logical host for this traffic. In the example configuration, “sc-qfs1” is the logical host name.

3. Build the shared host table on the metadata server.
   Use the Solaris Cluster `clnode show` command and syntax to obtain the host order information to build the table.
   For example:

```
ctelab30:root> clnode show

=== Cluster Nodes ===

Node Name:                                      ctelab30
  Node ID:                                         1
  Enabled:                                         yes
  privatehostname:                                 clusternode1-priv
  reboot_on_path_failure:                          disabled
  globalzoneshares:                                1
  defaultpsetmin:                                  1
  quorum_vote:                                     1
  quorum_defaultvote:                              1
  quorum_resv_key:                                 0x4AB78CC400000001
  Transport Adapter List:                          e1000g1, e1000g3

Node Name:                                      ctelab31
  Node ID:                                         2
  Enabled:                                         yes
  privatehostname:                                 clusternode2-priv
  reboot_on_path_failure:                          disabled
  globalzoneshares:                                1
  defaultpsetmin:                                  1
  quorum_vote:                                     1
  quorum_defaultvote:                              1
  quorum_resv_key:                                 0x4AB78CC400000002
  Transport Adapter List:                          e1000g1, e1000g3
```

Example **`/etc/opt/SUNWsamfs/hosts.Qfs1`** File

In the following example, each file system must have its own host table.

```
#
#MDS
# Shared MDS Host file for family set 'Qfs1'
#
#
ctelab30 clusternode1-priv,sc-qfs1 1 - server
ctelab31 clusternode2-priv,sc-qfs1 2 -
ctelab32 ctelab32,ctelab32-4 - -
ctelab33 ctelab33,ctelab33-4 - -
ctelab28 ctelab28,ctelab28-4 - -
```

Example **`/etc/opt/SUNWsamfs/hosts.Qfs1.local`** File

In the following example, each file system must have its own host table.

```
#
#MDS
# Local MDS Host file for family set 'Qfs1'
#
#
ctelab30 clusternode1-priv
ctelab31 clusternode2-priv
```

4. Create the file system by running `sammkfs` on each file system.

```
/opt/SUNWsamfs/sbin/sammkfs -S Qfs1
```

5. Add File System Mounts to `/etc/vfstab`
   Edit the `/etc/vfstab` file on each cluster node and append the Sun QFS file system mounts.

```
###
# QFS Filesystems
###
Qfs1 - /cluster/qfs1 samfs - no shared
Qfs2 - /cluster/qfs2 samfs - no shared
```

6. Mount the file systems.
   a. Make the the mount points on each cluster node.

   ```
   #mkdir -p /cluster/qfs1 /cluster/qfs2
   ```

   b. Mount Qfs1 and Qfs2 on each cluster node.
7. Create the Solaris Cluster metadata server resource group.
   When you have configured Solaris Cluster and Sun QFS, you need to create the MDS resource group under Solaris Cluster. Follow these steps:
   a. Add the Sun QFS resource type.

   ```
   # clresourcetype register SUNW.qfs
   ```

   b. Create a resource group for the metadata server.

   ```
   # clresourcegroup create -n ctelab30,ctelab31 sc-qfs-rg
   ```

   c. Add the logical host name to the resource group that you just created.

   ```
   # clreslogicalhostname create -g sc-qfs-rg \\
   -N qfs_ipmp1@ctelab30,qfs_ipmp1@ctelab31 sc-qfs1
   ```

   d. Add Sun QFS file system resources to the MDS resource group.

   ```
   # clresource create -g sc-qfs-rg -t SUNW.qfs \\
   -x QFSFileSystem=/cluster/qfs1,/cluster/qfs2 -y Resource_dependencies=sc-qfs1 \\
   fs-qfs-rs
   ```

   e. Bring the resource group online.

   ```
   # clresourcegroup online -emM sc-qfs-rg
   ```

   f. Check resource group status.

   ```
   # clresourcegroup status
   ```

8. Set local mode on the Sun QFS data devices.
   For each data device (`mr` device) that is part of the file system, set local mode.
   In the following example, each data device in the Qfs1 file system must have local mode set.

```
#/usr/cluster/bin/scconf -r -D name=dsk/d16,nodelist=ctelab31
#/usr/cluster/bin/scconf -c -D name=dsk/d16,localonly=true
#/usr/cluster/bin/scconf -r -D name=dsk/d10,nodelist=ctelab31
#/usr/cluster/bin/scconf -c -D name=dsk/d10,localonly=true
#/usr/cluster/bin/scconf -r -D name=dsk/d13,nodelist=ctelab31
#/usr/cluster/bin/scconf -c -D name=dsk/d13,localonly=true
```

How to Configure the File System Clients

After the operating system and Solaris Cluster software have been installed on the clients, install and configure the Sun QFS software on each metadata client.

1. Install the metadata client.
   Specific steps differ slightly, depending on the client architecture.
   - For Solaris systems
   a. Go to the location of the Sun QFS software packages.
      For example:

      ```
      # cd /net/eyelid/builds/src_ctl/R4_5/6/pkg/obj/SunOS_5.9_sparc
      ```

   b. Add the packages.

      ```
      pkgadd-d . SUNWqfsr SUNWqfsu
      ```

   c. If necessary, go to SunSolve and download any additional patches.
   d. If necessary, apply patches.
   - For Linux systems
     Follow similar steps to the Solaris process to install the base packages and apply any patches. Once the install completes, the install process will detect and build the `mcf` file configuration.
   a. Create mount directories and mount the packages as needed. For example:

      ```
      # mkdir /mnt/pkgs /mnt/pkg
      # mount eyelid-mn.central:/builds/src_ctl/R4_5/6/pkg-linux /mnt/pkgs
      # mount -o loop,ro iso_open.iso /mnt/pkg
      ```

   b. Add the packages.

      ```
      # /mnt/pkg/Install
      ```

   c. If necessary, go to SunSolve and download any additional patches.
   d. If necessary, apply patches.
2. Verify MPXIO has been enabled and clients can discover QFS devices.
3. Identify devices.
   Using the Solaris `format` command on the clients and the Solaris Cluster `cluster show -t device` command on the metadata server, match device ids that identify the devices that will be used for the Sun QFS configuration and build the `mcf` file on the Sparc clients.

   Example **`mcf`** File

```
#
# File system Qfs1
#
Qfs1 2 ma Qfs1 on shared
nodev 20 mm Qfs1 on
nodev 21 mm Qfs1 on
/dev/dsk/c6t600C0FF00000000000332B21D0B90000d0s0 22 mr Qfs1 on
/dev/dsk/c6t600C0FF0000000000876E9124FAF9C00d0s0 23 mr Qfs1 on
/dev/dsk/c6t600C0FF000000000004CAD7CC3CDE500d0s0 24 mr Qfs1 on
#
# File system Qfs2
#
Qfs2 5 ma Qfs2 on shared
nodev 50 mm Qfs2 on
nodev 51 mm Qfs2 on
/dev/dsk/c6t600C0FF00000000000332B057D2FF100d0s0 52 mr Qfs2 on
/dev/dsk/c6t600C0FF0000000000876E975EDA6A000d0s0 53 mr Qfs2 on
/dev/dsk/c6t600C0FF0000000000876E9780ECA8100d0s0 54 mr Qfs2 on
/dev/dsk/c6t600C0FF000000000004CAD139A855500d0s0 55 mr Qfs2 on
/dev/dsk/c6t600C0FF000000000004CAD4C40941C00d0s0 56 mr Qfs2 on
```

4. Configure the `/etc/opt/SUNWsamfs/defaults.conf` file on SPARC and Linux Clients.

```
trace
all = on
sam-fsd.size = 10M
sam-sharefsd.size = 10M
endtrace
```

5. Configure the `/etc/opt/SUNWsamfs/samfs.cmd` file on SPARC and Linux Clients.

```
fs = Qfs1
meta_timeo=0

fs = Qfs2
meta_timeo=0
```

6. Build the Sun QFS file system host tables.
   Using information already configued on the metadata server, build host tables. Follow the examples below.

> Note -
> Remember that for metadata communications between the metadata server and metadata clients, the clients that are not members of the cluster must communicate over the logical host.

Example **`/etc/opt/SUNWsamfs/hosts.Qfs1`** File

```
#
#MDC
# Shared Client Host file for family set 'Qfs1'
#
#
ctelab30 sc-qfs1 1 - server
ctelab31 sc-qfs1 2 -
ctelab32 ctelab32,ctelab32-4 - -
ctelab33 ctelab33,ctelab33-4 - -
ctelab28 ctelab28,ctelab28-4 - -
```

7. Bind each local host to the metadata server.
   Edit the local hosts file on each client to bind the file system to the metadata server.
   The `/etc/opt/SUNWsamfs/hosts.<fsname>.local` information will be different for each client. In the following example, the client is configured to use the interface on “ctelab32-4” to bind to host “sc-qfs1” for metadata traffic.

```
#
#MDC
# Local Client Host file for family set 'Qfs1'
#
#
ctelab30 sc-qfs1@ctelab32-4
ctelab31 sc-qfs1@ctelab32-4
```

8. Add file system mount information on each client.
   a. Edit the `/etc/vfstab` file on each client node.
   b. Append the QFS file system mounts.

   Example **`/etc/vfstab`** File (SPARC Systems Only)

```
###
# QFS Filesystems
###
Qfs1 - /cluster/qfs1 samfs - no shared
Qfs2 - /cluster/qfs2 samfs - no shared
```

   Example **`/etc/vfstab`** File (Linux Systems Only)

```
###
# QFS Fileystems
#
Qfs1 /cluster/qfs1 samfs noauto,rw,shared 0 0
Qfs2 /cluster/qfs2 samfs noauto,rw,shared 0 0
```

9. Mount file systems on each client.
   a. Make the the mount points on each client.

```
#mkdir -p /cluster/qfs1 /cluster/qfs2
```

   b. Mount the file systems on each cluster node.

```
#mount Qfs1;mount Qfs2
```

Configuration Example

The following example configuration uses shared and local file system host files for this Sun QFS shared file system setup. To communicate with metadata clients that are outside the Solaris Cluster, you must establish the Sun QFS metadata traffic over the Sun QFS network. Because the metadata client is not a member of the Solaris Cluster configuration, use a logical host for this traffic. In the example configuration, “sc-qfs1” is this host name.

The following example configuration files show a tested configuration that consists of four SPARC nodes and one AMD64 node. These nodes are identified as follows:

Name	Role	Architecture	Notes
`ctelab30`	Metadata server	SPARC	Solaris Cluster Node
`ctelab31`	Metadata server	SPARC	Solaris Cluster Node

ctelab32	Metadata client	SPARC	Sun QFS Client Node
ctelab33	Metadata client	SPARC	Sun QFS Client Node
ctelab28	Metadata client	AMD64	Sun QFS Linux Client Node

Sample **/etc/hosts** File for All Nodes

Prepare `/etc/hosts` file on all nodes to append settings to existing `/etc/hosts` table.

```
### SC Cluster Nodes ###
129.152.4.57 ctelab30 # Cluster Node
129.152.4.58 ctelab31 # Cluster Node
129.152.4.59 ctelab32 # QFS Client Node
129.152.4.60 ctelab33 # QFS Client Node
129.152.4.55 ctelab28 # QFS Client Node
### SC Logical ###
192.168.4.100 sc-qfs1
### QFS NET ###
## ctelab30
192.168.4.20 ctelab30-4
192.168.4.160 ctelab30-qfe1-test
192.168.4.210 ctelab30-qfe2-test
192.168.4.60 ctelab30-qfs
## ctelab31
192.168.4.21 ctelab31-4
192.168.4.161 ctelab31-qfe1-test
192.168.4.211 ctelab31-qfe2-test
192.168.4.61 ctelab31-qfs
## ctelab32
192.168.4.22 ctelab32-qfs
## ctelab33
192.168.4.23 ctelab33-qfs
## ctelab28
192.168.4.18 ctelab28-qfs
```

Sample Configuration Files for Metadata Server

/etc/nsswitch.conf File

Ensure that the `/etc/nsswitch.conf` includes cluster. For example:

```
hosts: cluster files dns nis
```

/etc/netmasks File

Add the subnet used in your `/etc/hosts` file to the `/etc/netmasks` file. For example:

```
192.168.4.0 255.255.255.0
```

/etc/hostname.qfe1 File

Identify the NIC port for `qfe1` in the `/etc/hostname.qfe1` file. For example:

```
ctelab30-4 netmask + broadcast + group qfs_ipmp1 up addif ctelab30-qfe1-test deprecated -failover
netmask + broadcast + up
```

/etc/hostname.qfe2 File

Identify the NIC port for `qfe2` in the `/etc/hostname.qfe2` file. For example:

```
ctelab30-qfe2-test netmask + broadcast + deprecated group qfs_ipmp1 -failover standby up
```

Client Network Configuration

In this example, both SPARC and AMD-64 clients are used. SPARC clients run the Solaris OS; AMD-64 clients use the Linux OS.

SPARC Solaris Client Network Configuration

- qfe0 used for Public network (129.152.4.0)
- qfe1 used for QFS private network. (192.168.4.0)

AMD-64 RedHat Linux Client Network Configuration

- bge0 used for Public network (129.152.4.0)
- bge1 used for QFS private network. (192.168.4.0)

> Note -
> On Linux OS, the NIC ports identified above are hardware address IDs and the OS identifies these as eth0,eth1 at the OS layer.

/etc/nsswitch.conf File (SPARC and Linux Clients)

For the clients, make sure that `cluster` is not listed in the `/etc/nsswitch.conf` file. For example:

```
hosts: files dns nis
```

/etc/hostname.qfe1 File (SPARC Clients Only)

• File should contain the following

```
ctelab32-4
```

/etc/sysconfig/network-scripts/ifcfg-eth1 File (Linux Clients Only)

• File should contain the following

```
DEVICE=eth1
BOOTPROTO=static
IPADDR=192.168.4.18
NETMASK=255.255.255.0
ONBOOT=yes
TYPE=Ethernet
```

Troubleshooting Issue With Oracle RAC

Problem: If you use Oracle RAC with Solaris Volume Manager for shared file systems, you might see a message that SVM volumes are not available.

Solution: In response to this issue, you can configure the `rac_maxdevretry` parameter in the `/etc/opt/SUNWsamfs/defaults.conf` file. By default, the `rac_maxdevretry` variable is set to 150 (5 minutes).

The items to take in account for adjusting this variable include:

- Number of nodes in the cluster
- Number of physical LUNs
- Numnber of SVM Diskgroups

- Speed of array topology 1G/2G/4G

In the following example `/etc/opt/SUNWsamfs/defaults.conf` file, the time is shortened to 90 (3 minutes).

```
trace
all = on
sam-sharefsd.options = all
sam-fsd.size = 10M
sam-sharefsd.size = 10M
endtrace
rac_maxdevretry = 90
```

This change was based on the following environment:

- Nodes - 8
- Physical LUNs - 32 Shared Physical LUNS w/4G Fibre Channel
- SVM disk groups - 1
- Shared file systems - 13

An example of this setting is included in `/etc/opt/SUNWsamfs/examples/defaults.conf`.

For more information about configuring an Oracle database, see the Sun Cluster Data Service for Oracle Real Application Clusters Guide for Solaris OS.

End of Complete (Printable) Guide to Using SAM-QFS With Solaris Cluster

Configuring Clients Outside of the Cluster

Configuring Clients Outside of the Cluster

You can configure a Sun QFS shared file system operating in a Solaris Cluster environment to support automated metadata server failover while providing file system access for clients that are not members of the Solaris Cluster configuration. This feature is commonly referred to as Clients Outside of The Cluster (COTC).

> Note -
> Before performing any COTC tasks, make sure that you are familiar with information about shared and local host tables.

COTC Task Map

You must complete the following procedures.

Step	Task	Description
1	Verify prerequisites and identify clients that are to be configured outside of the cluster.	See Configuration Prerequisites
2	Install Solaris Cluster software.	If not already done, install the Solaris Cluster software as explained in the Solaris Cluster information products.
3	Install Sun QFS software on the metadata server and shared clients.	See Installing Sun QFS.
4	Configure the metadata server.	See Configuring the Metadata Server.
5	Configure the clients.	See Configuring the File System Clients.

Configuration Requirements

The following table identifies specific configuration rules for each segment of the COTC environment:

Sun QFS Metadata Server	• mm devices must be on a separate physical LUN from data devices • mr devices must be on a separate physical LUN from metadata devices • mm devices must not have the Solaris Cluster "localonly" flag
Sun QFS Clients	• mm devices must be identified as `nodev` in the `mcf` file • `mr` devices must follow the `/dev/dsk` syntax, rather than the `/dev/did/dsk` syntax • The definition for the `mr` device on the client `/dev/dsk` must match the definition for the `/dev/did/dsk` on the metadata server
Solaris Cluster Devices	• Set "localonly" flag on `mr` devices
Private Metadata Network	• Solaris Cluster nodes that serve as metadata servers should use IPMP groups • Solaris Cluster Sun QFS resource groups should have a logical host name configured for metadata communication betweeen the metadata server and the clients that are outside of the cluster • Solaris Cluster Sun QFS resource groups should be defined to ensure that Sun QFS file system resources depend on the logical host name resource

How to Configure the Metadata Server

The following procedure includes embedded examples to better explain what you should expect to see in a COTC configuration.

1. Use the Solaris Cluster command `cluster show -t device` to identify the devices that will be used for the Sun QFS configuration and build the `mcf` file.
   Note the use of the `/dev/did` syntax in the following sample `mcf` file for this configuration.

```
#
# File system Qfs1
#
Qfs1 2 ma Qfs1 on shared
/dev/did/dsk/d7s0 20 mm Qfs1 on
/dev/did/dsk/d8s0 21 mm Qfs1 on
/dev/did/dsk/d16s0 22 mr Qfs1 on
/dev/did/dsk/d10s0 23 mr Qfs1 on
/dev/did/dsk/d13s0 24 mr Qfs1 on
#
# File system Qfs2
#
Qfs2 5 ma Qfs2 on shared
/dev/did/dsk/d9s0 50 mm Qfs2 on
/dev/did/dsk/d11s0 51 mm Qfs2 on
/dev/did/dsk/d17s0 52 mr Qfs2 on
/dev/did/dsk/d12s0 53 mr Qfs2 on
/dev/did/dsk/d14s0 54 mr Qfs2 on
/dev/did/dsk/d15s0 55 mr Qfs2 on
/dev/did/dsk/d18s0 56 mr Qfs2 on
```

Example **`/etc/opt/SUNWsamfs/defaults.conf`** File

```
trace
all = on
sam-fsd.size = 10M
sam-sharefsd.size = 10M
endtrace
```

Example **`/etc/opt/SUNWsamfs/samfs.cmd`** File

```
fs = Qfs1
meta_timeo=0

fs = Qfs2
meta_timeo=0
```

2. Build the Sun QFS file system hosts tables.
   The example configuration uses shared and local file system host files for this QFS shared file system setup.

> Note -
> To communicate with metadata clients that are outside of the cluster, you must establish Sun QFS metadata traffic over the Sun QFS network. Because the metadata client is not a member of the Solaris Cluster configuration, the example uses a logical host for this traffic. In the example configuration, “sc-qfs1” is the logical host name.

3. Build the shared host table on the metadata server.
   Use the Solaris Cluster `clnode show` command and syntax to obtain the host order information to build the table.
   For example:

```
ctelab30:root> clnode show

=== Cluster Nodes ===

Node Name:                                      ctelab30
  Node ID:                                         1
  Enabled:                                         yes
  privatehostname:                                 clusternode1-priv
  reboot_on_path_failure:                          disabled
  globalzoneshares:                                1
  defaultpsetmin:                                  1
  quorum_vote:                                     1
  quorum_defaultvote:                              1
  quorum_resv_key:                                 0x4AB78CC400000001
  Transport Adapter List:                          e1000g1, e1000g3

Node Name:                                      ctelab31
  Node ID:                                         2
  Enabled:                                         yes
  privatehostname:                                 clusternode2-priv
  reboot_on_path_failure:                          disabled
  globalzoneshares:                                1
  defaultpsetmin:                                  1
  quorum_vote:                                     1
  quorum_defaultvote:                              1
  quorum_resv_key:                                 0x4AB78CC400000002
  Transport Adapter List:                          e1000g1, e1000g3
```

Example **/etc/opt/SUNWsamfs/hosts.Qfs1** File

In the following example, each file system must have its own host table.

```
#
#MDS
# Shared MDS Host file for family set 'Qfs1'
#
#
ctelab30 clusternode1-priv,sc-qfs1 1 - server
ctelab31 clusternode2-priv,sc-qfs1 2 -
ctelab32 ctelab32,ctelab32-4 - -
ctelab33 ctelab33,ctelab33-4 - -
ctelab28 ctelab28,ctelab28-4 - -
```

Example **`/etc/opt/SUNWsamfs/hosts.Qfs1.local`** File

In the following example, each file system must have its own host table.

```
#
#MDS
# Local MDS Host file for family set 'Qfs1'
#
#
ctelab30 clusternode1-priv
ctelab31 clusternode2-priv
```

4. Create the file system by running `sammkfs` on each file system.

```
/opt/SUNWsamfs/sbin/sammkfs -S Qfs1
```

5. Add File System Mounts to `/etc/vfstab`
   Edit the `/etc/vfstab` file on each cluster node and append the Sun QFS file system mounts.

```
###
# QFS Filesystems
###
Qfs1 - /cluster/qfs1 samfs - no shared
Qfs2 - /cluster/qfs2 samfs - no shared
```

6. Mount the file systems.
   a. Make the the mount points on each cluster node.

   ```
   #mkdir -p /cluster/qfs1 /cluster/qfs2
   ```

   b. Mount Qfs1 and Qfs2 on each cluster node.
7. Create the Solaris Cluster metadata server resource group.
   When you have configured Solaris Cluster and Sun QFS, you need to create the MDS resource group under Solaris Cluster. Follow these steps:
   a. Add the Sun QFS resource type.

   ```
   # clresourcetype register SUNW.qfs
   ```

   b. Create a resource group for the metadata server.

   ```
   # clresourcegroup create -n ctelab30,ctelab31 sc-qfs-rg
   ```

   c. Add the logical host name to the resource group that you just created.

   ```
   # clreslogicalhostname create -g sc-qfs-rg \\
   -N qfs_ipmp1@ctelab30,qfs_ipmp1@ctelab31 sc-qfs1
   ```

   d. Add Sun QFS file system resources to the MDS resource group.

```
# clresource create -g sc-qfs-rg -t SUNW.qfs \\
-x QFSFileSystem=/cluster/qfs1,/cluster/qfs2 -y Resource_dependencies=sc-qfs1 \\
fs-qfs-rs
```

e. Bring the resource group online.

```
# clresourcegroup online -emM sc-qfs-rg
```

f. Check resource group status.

```
# clresourcegroup status
```

8. Set local mode on the Sun QFS data devices.
For each data device (`mr` device) that is part of the file system, set local mode.
In the following example, each data device in the Qfs1 file system must have local mode set.

```
#/usr/cluster/bin/scconf -r -D name=dsk/d16,nodelist=ctelab31
#/usr/cluster/bin/scconf -c -D name=dsk/d16,localonly=true
#/usr/cluster/bin/scconf -r -D name=dsk/d10,nodelist=ctelab31
#/usr/cluster/bin/scconf -c -D name=dsk/d10,localonly=true
#/usr/cluster/bin/scconf -r -D name=dsk/d13,nodelist=ctelab31
#/usr/cluster/bin/scconf -c -D name=dsk/d13,localonly=true
```

How to Configure the File System Clients

After the operating system and Solaris Cluster software have been installed on the clients, install and configure the Sun QFS software on each metadata client.

1. Install the metadata client.
Specific steps differ slightly, depending on the client architecture.
   - For Solaris systems
   a. Go to the location of the Sun QFS software packages.
   For example:

```
# cd /net/eyelid/builds/src_ctl/R4_5/6/pkg/obj/SunOS_5.9_sparc
```

   b. Add the packages.

```
pkgadd-d . SUNWqfsr SUNWqfsu
```

   c. If necessary, go to SunSolve and download any additional patches.
   d. If necessary, apply patches.
   - For Linux systems
   Follow similar steps to the Solaris process to install the base packages and apply any patches. Once the install completes, the install process will detect and build the `mcf` file configuration.
   a. Create mount directories and mount the packages as needed. For example:

```
# mkdir /mnt/pkgs /mnt/pkg
# mount eyelid-mn.central:/builds/src_ctl/R4_5/6/pkg-linux /mnt/pkgs
# mount -o loop,ro iso_open.iso /mnt/pkg
```

   b. Add the packages.

```
# /mnt/pkg/Install
```

   c. If necessary, go to SunSolve and download any additional patches.
   d. If necessary, apply patches.
2. Verify MPXIO has been enabled and clients can discover QFS devices.
3. Identify devices.
   Using the Solaris `format` command on the clients and the Solaris Cluster `cluster show -t device` command on the metadata server, match device ids that identify the devices that will be used for the Sun QFS configuration and build the `mcf` file on the Sparc clients.

   Example **`mcf`** File

```
#
# File system Qfs1
#
Qfs1 2 ma Qfs1 on shared
nodev 20 mm Qfs1 on
nodev 21 mm Qfs1 on
/dev/dsk/c6t600C0FF00000000000332B21D0B90000d0s0 22 mr Qfs1 on
/dev/dsk/c6t600C0FF0000000000876E9124FAF9C00d0s0 23 mr Qfs1 on
/dev/dsk/c6t600C0FF000000000004CAD7CC3CDE500d0s0 24 mr Qfs1 on
#
# File system Qfs2
#
Qfs2 5 ma Qfs2 on shared
nodev 50 mm Qfs2 on
nodev 51 mm Qfs2 on
/dev/dsk/c6t600C0FF00000000000332B057D2FF100d0s0 52 mr Qfs2 on
/dev/dsk/c6t600C0FF0000000000876E975EDA6A000d0s0 53 mr Qfs2 on
/dev/dsk/c6t600C0FF0000000000876E9780ECA8100d0s0 54 mr Qfs2 on
/dev/dsk/c6t600C0FF000000000004CAD139A855500d0s0 55 mr Qfs2 on
/dev/dsk/c6t600C0FF000000000004CAD4C40941C00d0s0 56 mr Qfs2 on
```

4. Configure the `/etc/opt/SUNWsamfs/defaults.conf` file on SPARC and Linux Clients.

```
trace
all = on
sam-fsd.size = 10M
sam-sharefsd.size = 10M
endtrace
```

5. Configure the `/etc/opt/SUNWsamfs/samfs.cmd` file on SPARC and Linux Clients.

```
fs = Qfs1
meta_timeo=0

fs = Qfs2
meta_timeo=0
```

6. Build the Sun QFS file system host tables.
   Using information already configued on the metadata server, build host tables. Follow the examples below.

> Note -
> Remember that for metadata communications between the metadata server and metadata clients, the clients that are not members of the cluster must communicate over the logical host.

Example **`/etc/opt/SUNWsamfs/hosts.Qfs1`** File

```
#
#MDC
# Shared Client Host file for family set 'Qfs1'
#
#
ctelab30 sc-qfs1 1 - server
ctelab31 sc-qfs1 2 -
ctelab32 ctelab32,ctelab32-4 - -
ctelab33 ctelab33,ctelab33-4 - -
ctelab28 ctelab28,ctelab28-4 - -
```

7. Bind each local host to the metadata server.
   Edit the local hosts file on each client to bind the file system to the metadata server.
   The `/etc/opt/SUNWsamfs/hosts.<fsname>.local` information will be different for each client. In the following example, the client is configured to use the interface on "ctelab32-4" to bind to host "sc-qfs1" for metadata traffic.

```
#
#MDC
# Local Client Host file for family set 'Qfs1'
#
#
ctelab30 sc-qfs1@ctelab32-4
ctelab31 sc-qfs1@ctelab32-4
```

8. Add file system mount information on each client.
   a. Edit the `/etc/vfstab` file on each client node.
   b. Append the QFS file system mounts.

      Example **`/etc/vfstab`** File (SPARC Systems Only)

```
###
# QFS Filesystems
###
Qfs1 - /cluster/qfs1 samfs - no shared
Qfs2 - /cluster/qfs2 samfs - no shared
```

      Example **`/etc/vfstab`** File (Linux Systems Only)

```
###
# QFS Fileystems
#
Qfs1 /cluster/qfs1 samfs noauto,rw,shared 0 0
Qfs2 /cluster/qfs2 samfs noauto,rw,shared 0 0
```

9. Mount file systems on each client.
   a. Make the the mount points on each client.

```
#mkdir -p /cluster/qfs1 /cluster/qfs2
```

   b. Mount the file systems on each cluster node.

```
#mount Qfs1;mount Qfs2
```

Configuration Example

The following example configuration uses shared and local file system host files for this Sun QFS shared file system setup. To communicate with metadata clients that are outside the Solaris Cluster, you must establish the Sun QFS metadata traffic over the Sun QFS network. Because the metadata client is not a member of the Solaris Cluster configuration, use a logical host for this traffic. In the example configuration, “sc-qfs1” is this host name.

The following example configuration files show a tested configuration that consists of four SPARC nodes and one AMD64 node. These nodes are identified as follows:

Name	Role	Architecture	Notes
`ctelab30`	Metadata server	SPARC	Solaris Cluster Node
`ctelab31`	Metadata server	SPARC	Solaris Cluster Node
`ctelab32`	Metadata client	SPARC	Sun QFS Client Node
`ctelab33`	Metadata client	SPARC	Sun QFS Client Node
`ctelab28`	Metadata client	AMD64	Sun QFS Linux Client Node

Sample **`/etc/hosts`** File for All Nodes

Prepare `/etc/hosts` file on all nodes to append settings to existing `/etc/hosts` table.

```
### SC Cluster Nodes ###
129.152.4.57 ctelab30 # Cluster Node
129.152.4.58 ctelab31 # Cluster Node
129.152.4.59 ctelab32 # QFS Client Node
129.152.4.60 ctelab33 # QFS Client Node
129.152.4.55 ctelab28 # QFS Client Node
### SC Logical ###
192.168.4.100 sc-qfs1
### QFS NET ###
## ctelab30
192.168.4.20 ctelab30-4
192.168.4.160 ctelab30-qfe1-test
192.168.4.210 ctelab30-qfe2-test
192.168.4.60 ctelab30-qfs
## ctelab31
192.168.4.21 ctelab31-4
192.168.4.161 ctelab31-qfe1-test
192.168.4.211 ctelab31-qfe2-test
192.168.4.61 ctelab31-qfs
## ctelab32
192.168.4.22 ctelab32-qfs
## ctelab33
192.168.4.23 ctelab33-qfs
## ctelab28
192.168.4.18 ctelab28-qfs
```

Sample Configuration Files for Metadata Server

`/etc/nsswitch.conf` File

Ensure that the `/etc/nsswitch.conf` includes cluster. For example:

```
hosts: cluster files dns nis
```

`/etc/netmasks` File

Add the subnet used in your `/etc/hosts` file to the `/etc/netmasks` file. For example:

```
192.168.4.0 255.255.255.0
```

/etc/hostname.qfe1 File

Identify the NIC port for `qfe1` in the `/etc/hostname.qfe1` file. For example:

```
ctelab30-4 netmask + broadcast + group qfs_ipmp1 up addif ctelab30-qfe1-test deprecated -failover
netmask + broadcast + up
```

/etc/hostname.qfe2 File

Identify the NIC port for `qfe2` in the `/etc/hostname.qfe2` file. For example:

```
ctelab30-qfe2-test netmask + broadcast + deprecated group qfs_ipmp1 -failover standby up
```

Client Network Configuration

In this example, both SPARC and AMD-64 clients are used. SPARC clients run the Solaris OS; AMD-64 clients use the Linux OS.

SPARC Solaris Client Network Configuration

- qfe0 used for Public network (129.152.4.0)
- qfe1 used for QFS private network. (192.168.4.0)

AMD-64 RedHat Linux Client Network Configuration

- bge0 used for Public network (129.152.4.0)
- bge1 used for QFS private network. (192.168.4.0)

> Note -
> On Linux OS, the NIC ports identified above are hardware address IDs and the OS identifies these as eth0,eth1 at the OS layer.

/etc/nsswitch.conf File (SPARC and Linux Clients)

For the clients, make sure that `cluster` is not listed in the `/etc/nsswitch.conf` file. For example:

```
hosts: files dns nis
```

/etc/hostname.qfe1 File (SPARC Clients Only)

• File should contain the following

```
ctelab32-4
```

/etc/sysconfig/network-scripts/ifcfg-eth1 File (Linux Clients Only)

• File should contain the following

```
DEVICE=eth1
BOOTPROTO=static
IPADDR=192.168.4.18
NETMASK=255.255.255.0
ONBOOT=yes
TYPE=Ethernet
```

COTC Example - Client in Cluster

Example Configuration for SPARC Client in Cluster

> **Note -**
> Compare this configuration with COTC Example - Client Not in Cluster.

```
lake9-mn% more /etc/opt/SUNWsamfs/mcf
# SAM-QFS file system
# Equipment             Eq      Eq      Family  Device  Additional
# Identifier            Ord     Type    Set     State   Parameters
# -----------           ---     ----    ------  ------  ----------
samfs1                  1       ma      samfs1  on shared
/dev/did/dsk/d6s0 11      mm      samfs1  on      /dev/did/rdsk/d6s0
/dev/did/dsk/d5s0 12 md samfs1 on /dev/did/rdsk/d5s0

samfs2                  2       ma      samfs2  on      shared
/dev/did/dsk/d6s1       21      mm      samfs2  on      /dev/did/rdsk/d6s1
/dev/did/dsk/d4s0       22      md      samfs2  on      /dev/did/rdsk/d4s0

samfs3                  3       ma      samfs3  on      shared
/dev/did/dsk/d6s2       31      mm      samfs3  on      /dev/did/rdsk/d6s2
/dev/did/dsk/d3s0       32      md      samfs3  on      /dev/did/rdsk/d3s0

lake9-mn% more /etc/hosts
#
# Internet host table
#
::1 localhost
127.0.0.1 localhost
10.1.202.55 lake9-mn loghost
10.1.202.11 eagan-mn
10.1.202.225 lake10-mn # Cluster Node
10.1.202.54 lake8-mn # Cluster Node
10.1.202.53 lake7-mn # Cluster Node

lake9-mn% more /etc/opt/SUNWsamfs/hosts.samfs*
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs1
::::::::::::::
lake7-mn lake7-mn 1 0 server
lake8-mn lake8-mn 1 0
lake9-mn lake9-mn 1 0
lake10-mn lake10-mn 1 0
teacup-mn teacup-mn - -
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs1.local
::::::::::::::
lake7-mn clusternode4-priv
lake8-mn clusternode3-priv
lake9-mn clusternode2-priv
lake10-mn clusternode1-priv
teacup-mn teacup-mn - -
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs2
::::::::::::::
lake7-mn lake7-mn 1 0 server
lake8-mn lake8-mn 1 0
lake9-mn lake9-mn 1 0
lake10-mn lake10-mn 1 0
teacup-mn teacup-mn - -
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs2.local
::::::::::::::
lake7-mn clusternode4-priv
lake8-mn clusternode3-priv
lake9-mn clusternode2-priv
lake10-mn clusternode1-priv
teacup-mn teacup-mn - -
```

```
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs3
::::::::::::::
lake7-mn lake7-mn 1 0 server
lake8-mn lake8-mn 1 0
lake9-mn lake9-mn 1 0
lake10-mn lake10-mn 1 0
teacup-mn teacup-mn - -
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs3.local
::::::::::::::
lake7-mn clusternode4-priv
lake8-mn clusternode3-priv
lake9-mn clusternode2-priv
lake10-mn clusternode1-priv
teacup-mn teacup-mn - -

lake9-mn% more /etc/vfstab
#device device mount FS fsck mount mount
#to mount to fsck  point  type pass at boot options
#
fd - /dev/fd fd - no -
/proc - /proc proc - no -
/dev/dsk/c0t0d0s1 - - swap - no -
/dev/dsk/c0t0d0s3 /dev/rdsk/c0t0d0s3 / ufs 1 no logging
/dev/dsk/c0t0d0s0 /dev/rdsk/c0t0d0s0 /sol11 ufs 2 yes -
#/dev/dsk/c0t1d0s2 /dev/rdsk/c0t1d0s2 /globaldevices ufs      -       yes     -
/devices - /devices devfs - no -
sharefs - /etc/dfs/sharetab sharefs - no -
ctfs - /system/contract ctfs - no -
objfs - /system/object objfs - no -
swap - /tmp tmpfs - yes -
/dev/did/dsk/d35s2 /dev/did/rdsk/d35s2 /global/.devices/node@2 ufs 2 no global
samfs1 - /sam1 samfs - no shared,wr_throttle=2048
samfs2 - /sam2 samfs - no shared,wr_throttle=2048
samfs3 - /sam3 samfs - no shared,wr_throttle=2048

lake9-mn% more /etc/nsswitch.conf
#
# /etc/nsswitch.nis:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# uses NIS (YP) in conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet" transports.

# NIS service requires that svc:/network/nis/client:default be enabled
# and online.

# the following two lines obviate the "+" entry in /etc/passwd and /etc/group.
passwd:     files nis
group:      files nis

# consult /etc "files" only if nis is down.
#hosts: files nis dns
hosts: cluster files nis dns

# Note that IPv4 addresses are searched for in all of the ipnodes databases
# before searching the hosts databases.
#ipnodes:    nis [NOTFOUND=return] files
ipnodes:    files nis [TRYAGAIN=0 NOTFOUND=return]

#networks:   nis [NOTFOUND=return] files
networks:   files nis [NOTFOUND=return]
#protocols:  nis [NOTFOUND=return] files
protocols:  files nis [NOTFOUND=return]
#rpc:         nis [NOTFOUND=return] files
rpc:        files nis [NOTFOUND=return]
#ethers:      nis [NOTFOUND=return] files
ethers:     files nis [NOTFOUND=return]
#netmasks:    nis [NOTFOUND=return] files
netmasks:   cluster files nis [NOTFOUND=return]
#bootparams: nis [NOTFOUND=return] files
bootparams: files nis [NOTFOUND=return]
```

```
#publickey:  nis [NOTFOUND=return] files
publickey:  files nis [NOTFOUND=return]

netgroup:   nis

automount:  files nis
aliases:    files nis

# for efficient getservbyname() avoid nis
services:   files nis
printers: user files nis

auth_attr:  files nis
prof_attr:  files nis
project:    files nis
```

COTC Example - Client Not in Cluster

Example Configuration for SPARC Client Not in Cluster

```
teacup-mn% more /etc/opt/SUNWsamfs/mcf
# SAM-QFS file system
# Equipment             Eq      Eq      Family  Device  Additional
# Identifier            Ord     Type    Set     State   Parameters
# -----------           ---     ----    ------  ------  ----------
samfs1                  1       ma      samfs1  on shared
nodev 11      mm      samfs1  on
/dev/dsk/c3t600A0B80002A24F200000BE24A38E08Cd0s0 12 md samfs1 on
/dev/rdsk/c3t600A0B80002A24F200000BE24A38E08Cd0s0

samfs2                  2       ma      samfs2  on      shared
nodev       21      mm      samfs2  on
/dev/dsk/c3t600A0B80002A24D80000053C4A38E2F1d0s0       22      md      samfs2  on
/dev/rdsk/c3t600A0B80002A24D80000053C4A38E2F1d0s0

samfs3                  3       ma      samfs3  on      shared
nodev       31      mm      samfs3  on
/dev/dsk/c3t600A0B80002A24F200000BFF4A38E16Ad0s0       32      md      samfs3  on
/dev/rdsk/c3t600A0B80002A24F200000BFF4A38E16Ad0s0

teacup-mn% more /etc/hosts
#
# Internet host table
#
::1 localhost
127.0.0.1 localhost
10.1.201.162 teacup-mn loghost
10.1.202.11 eagan-mn
clusternode4-priv lake7-mn
clusternode3-priv lake8-mn
clusternode2-priv lake9-mn
clusternode1-priv lake10-mn

teacup-mn% more /etc/opt/SUNWsamfs/hosts.samfs*
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs1
::::::::::::::
lake7-mn lake7-mn 1 0 server
lake8-mn lake8-mn 1 0
lake9-mn lake9-mn 1 0
lake10-mn lake10-mn 1 0
teacup-mn teacup-mn - -
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs1.local
::::::::::::::
lake7-mn lake7-mn
lake8-mn lake8-mn
lake9-mn lake9-mn
lake10-mn lake10-mn
teacup-mn teacup-mn
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs2
::::::::::::::
lake7-mn lake7-mn 1 0 server
lake8-mn lake8-mn 1 0
lake9-mn lake9-mn 1 0
lake10-mn lake10-mn 1 0
teacup-mn teacup-mn - -
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs2.local
::::::::::::::
```

```
lake7-mn lake7-mn
lake8-mn lake8-mn
lake9-mn lake9-mn
lake10-mn lake10-mn
teacup-mn teacup-mn
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs3
::::::::::::::
lake7-mn lake7-mn 1 0 server
lake8-mn lake8-mn 1 0
lake9-mn lake9-mn 1 0
lake10-mn lake10-mn 1 0
teacup-mn teacup-mn - -
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs3.local
::::::::::::::
lake7-mn lake7-mn
lake8-mn lake8-mn
lake9-mn lake9-mn
lake10-mn lake10-mn
teacup-mn teacup-mn

teacup-mn% more /etc/vfstab
#device  device  mount  FS fsck mount mount
#to mount to fsck  point  type pass at boot options
#
fd - /dev/fd fd - no -
/proc - /proc proc - no -
/dev/dsk/c0t0d0s1 - - swap - no -
/dev/dsk/c0t0d0s0 /dev/rdsk/c0t0d0s0 / ufs 1 no logging
/devices - /devices devfs - no -
sharefs - /etc/dfs/sharetab sharefs - no -
ctfs - /system/contract ctfs - no -
objfs - /system/object objfs - no -
swap - /tmp tmpfs - yes -
#s1 - /s1 samfs - no trace,shared,noclustermgmt
#s3 - /s3 samfs - no nologging,trace,nosam
#s4 - /s4 samfs - no trace,shared
#s5 - /s5 samfs - no nologging,trace,nosam
samfs1  -       /sam1 samfs   -       no      shared
samfs2  -       /sam2 samfs   -       no      shared
samfs3  -       /sam3 samfs   -       no      shared

teacup-mn% more /etc/nsswitch.conf
#
# /etc/nsswitch.nis:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# uses NIS (YP) in conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet" transports.

# NIS service requires that svc:/network/nis/client:default be enabled
# and online.

# the following two lines obviate the "+" entry in /etc/passwd and /etc/group.
passwd:     files nis
group:      files nis

# consult /etc "files" only if nis is down.
hosts: files nis dns

# Note that IPv4 addresses are searched for in all of the ipnodes databases
# before searching the hosts databases.
ipnodes:    nis [NOTFOUND=return] files

networks:   nis [NOTFOUND=return] files
protocols:  nis [NOTFOUND=return] files
rpc:        nis [NOTFOUND=return] files
ethers:     nis [NOTFOUND=return] files
netmasks:   nis [NOTFOUND=return] files
bootparams: nis [NOTFOUND=return] files
publickey:  nis [NOTFOUND=return] files
```

```
netgroup:   nis

automount:  files nis
aliases:    files nis

# for efficient getservbyname() avoid nis
services:   files nis
printers: user files nis

auth_attr:  files nis
prof_attr:  files nis
project:    files nis
```

COTC Example - Metadata Server

Example Configuration for Metadata Server for a Cluster

```
lake7-mn% more /etc/opt/SUNWsamfs/mcf
# SAM-QFS file system
# Equipment             Eq      Eq      Family  Device  Additional
# Identifier            Ord     Type    Set     State   Parameters
# -----------           ---     ----    ------  ------  ----------
samfs1                  1       ma      samfs1  on shared
/dev/did/dsk/d6s0 11      mm      samfs1  on      /dev/did/rdsk/d6s0
/dev/did/dsk/d5s0 12 md samfs1 on /dev/did/rdsk/d5s0

samfs2                  2       ma      samfs2  on      shared
/dev/did/dsk/d6s1       21      mm      samfs2  on      /dev/did/rdsk/d6s1
/dev/did/dsk/d4s0       22      md      samfs2  on      /dev/did/rdsk/d4s0

samfs3                  3       ma      samfs3  on      shared
/dev/did/dsk/d6s2       31      mm      samfs3  on      /dev/did/rdsk/d6s2
/dev/did/dsk/d3s0       32      md      samfs3  on      /dev/did/rdsk/d3s0

lake7-mn% more /etc/hosts
#
# Internet host table
#
::1 localhost
127.0.0.1 localhost
10.1.202.53 lake7-mn loghost
10.1.202.11 eagan-mn
10.1.202.225 lake10-mn # Cluster Node
10.1.202.55 lake9-mn # Cluster Node
10.1.202.54 lake8-mn # Cluster Node

lake7-mn% more /etc/opt/SUNWsamfs/hosts.samfs*
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs1
::::::::::::::
lake7-mn lake7-mn,clusternode4-priv 1 0 server
lake8-mn lake8-mn,clusternode3-priv 1 0
lake9-mn lake9-mn,clusternode2-priv 1 0
lake10-mn lake10-mn,clusternode1-priv 1 0
teacup-mn teacup-mn - -
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs1.local
::::::::::::::
lake7-mn clusternode4-priv
lake8-mn clusternode3-priv
lake9-mn clusternode2-priv
lake10-mn clusternode1-priv
teacup-mn teacup-mn
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs2
::::::::::::::
lake7-mn lake7-mn,clusternode4-priv 1 0 server
lake8-mn lake8-mn,clusternode3-priv 1 0
lake9-mn lake9-mn,clusternode2-priv 1 0
lake10-mn lake10-mn,clusternode1-priv 1 0
teacup-mn teacup-mn - -
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs2.local
::::::::::::::
lake7-mn clusternode4-priv
lake8-mn clusternode3-priv
lake9-mn clusternode2-priv
lake10-mn clusternode1-priv
```

```
teacup-mn teacup-mn
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs3
::::::::::::::
lake7-mn lake7-mn,clusternode4-priv 1 0 server
lake8-mn lake8-mn,clusternode3-priv 1 0
lake9-mn lake9-mn,clusternode2-priv 1 0
lake10-mn lake10-mn,clusternode1-priv 1 0
teacup-mn teacup-mn - -
::::::::::::::
/etc/opt/SUNWsamfs/hosts.samfs3.local
::::::::::::::
lake7-mn clusternode4-priv
lake8-mn clusternode3-priv
lake9-mn clusternode2-priv
lake10-mn clusternode1-priv
teacup-mn teacup-mn

lake7-mn% more /etc/vfstab
#device device mount FS fsck mount mount
#to mount to fsck  point  type pass at boot options
#
fd - /dev/fd fd - no -
/proc - /proc proc - no -
/dev/dsk/c0t0d0s1 - - swap - no -
/dev/dsk/c0t0d0s3 /dev/rdsk/c0t0d0s3 / ufs 1 no logging
/dev/dsk/c0t0d0s0 /dev/rdsk/c0t0d0s0 /sol11 ufs 2 yes -
#/dev/dsk/c0t1d0s2 /dev/rdsk/c0t1d0s2 /globaldevices ufs      -       yes     -
/devices - /devices devfs - no -
sharefs - /etc/dfs/sharetab sharefs - no -
ctfs - /system/contract ctfs - no -
objfs - /system/object objfs - no -
swap - /tmp tmpfs - yes -
/dev/did/dsk/d43s2 /dev/did/rdsk/d43s2 /global/.devices/node@4 ufs 2 no global
samfs1  -       /sam1 samfs   -       no      shared
samfs2  -       /sam2 samfs   -       no      shared
samfs3  -       /sam3 samfs   -       no      shared

lake7-mn% more /etc/nsswitch.conf
#
# /etc/nsswitch.nis:
#
# An example file that could be copied over to /etc/nsswitch.conf; it
# uses NIS (YP) in conjunction with files.
#
# "hosts:" and "services:" in this file are used only if the
# /etc/netconfig file has a "-" for nametoaddr_libs of "inet" transports.

# NIS service requires that svc:/network/nis/client:default be enabled
# and online.

# the following two lines obviate the "+" entry in /etc/passwd and /etc/group.
passwd:     files nis
group:      files nis

# consult /etc "files" only if nis is down.
#hosts: files nis dns
hosts: cluster files nis dns

# Note that IPv4 addresses are searched for in all of the ipnodes databases
# before searching the hosts databases.
#ipnodes:    nis [NOTFOUND=return] files
ipnodes:    files nis [TRYAGAIN=0 NOTFOUND=return]

#networks:   nis [NOTFOUND=return] files
networks:   files nis [NOTFOUND=return]
#protocols:  nis [NOTFOUND=return] files
protocols:  files nis [NOTFOUND=return]
#rpc:         nis [NOTFOUND=return] files
rpc:        files nis [NOTFOUND=return]
#ethers:      nis [NOTFOUND=return] files
ethers:     files nis [NOTFOUND=return]
#netmasks:   nis [NOTFOUND=return] files
netmasks:   cluster files nis [NOTFOUND=return]
#bootparams: nis [NOTFOUND=return] files
```

```
bootparams: files nis [NOTFOUND=return]
#publickey:  nis [NOTFOUND=return] files
publickey:  files nis [NOTFOUND=return]

netgroup:   nis

automount:  files nis
aliases:    files nis

# for efficient getservbyname() avoid nis
services:   files nis
printers: user files nis

auth_attr:  files nis
prof_attr:  files nis
project:    files nis

lake7-mn% cluster show -t dev

=== DID Device Instances ===

DID Device Name:                                /dev/did/rdsk/d1
  Full Device Path:                                lake10-mn:/dev/rdsk/c0t16d0
  Replication:                                     none
  default_fencing:                                 global

DID Device Name:                                /dev/did/rdsk/d2
  Full Device Path:                                lake10-mn:/dev/rdsk/c0t17d0
  Replication:                                     none
  default_fencing:                                 global

DID Device Name:                                /dev/did/rdsk/d3
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A24F200000BFF4A38E16Ad0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A24F200000BFF4A38E16Ad0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A24F200000BFF4A38E16Ad0
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A24F200000BFF4A38E16Ad0
  Replication:                                     none
  default_fencing:                                 nofencing

DID Device Name:                                /dev/did/rdsk/d4
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A24D80000053C4A38E2F1d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A24D80000053C4A38E2F1d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A24D80000053C4A38E2F1d0
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A24D80000053C4A38E2F1d0
  Replication:                                     none
  default_fencing:                                 nofencing

DID Device Name:                                /dev/did/rdsk/d5
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A24F200000BE24A38E08Cd0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A24F200000BE24A38E08Cd0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A24F200000BE24A38E08Cd0
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A24F200000BE24A38E08Cd0
  Replication:                                     none
  default_fencing:                                 nofencing

DID Device Name:                                /dev/did/rdsk/d6
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A24D80000053A4A38E1FBd0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A24D80000053A4A38E1FBd0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A24D80000053A4A38E1FBd0
  Full Device Path:
```

```
lake8-mn:/dev/rdsk/c6t600A0B80002A24D80000053A4A38E1FBd0
  Replication:                                   none
  default_fencing:                               global

DID Device Name:                               /dev/did/rdsk/d7
  Full Device Path:                                 lake8-mn:/dev/rdsk/c4t202400A0B82A271Ed31
  Full Device Path:                                 lake10-mn:/dev/rdsk/c1t202400A0B82A271Ed31
  Full Device Path:                                 lake9-mn:/dev/rdsk/c4t202400A0B82A271Ed31
  Full Device Path:                                 lake7-mn:/dev/rdsk/c4t202400A0B82A271Ed31
  Replication:                                   none
  default_fencing:                               nofencing

DID Device Name:                               /dev/did/rdsk/d8
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A27180000052F471891E4d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A27180000052F471891E4d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A27180000052F471891E4d0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A27180000052F471891E4d0
  Replication:                                   none
  default_fencing:                               global

DID Device Name:                               /dev/did/rdsk/d9
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A27180000052B471891B4d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A27180000052B471891B4d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A27180000052B471891B4d0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A27180000052B471891B4d0
  Replication:                                   none
  default_fencing:                               global

DID Device Name:                               /dev/did/rdsk/d10
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A271800000052747189182d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A271800000052747189182d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A271800000052747189182d0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A271800000052747189182d0
  Replication:                                   none
  default_fencing:                               global

DID Device Name:                               /dev/did/rdsk/d11
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A2718000000523471891 4Cd0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A27180000005234718914Cd0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A27180000005234718914Cd0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A27180000005234718914Cd0
  Replication:                                   none
  default_fencing:                               global

DID Device Name:                               /dev/did/rdsk/d12
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A27180000051F4718911Cd0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A27180000051F4718911Cd0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A27180000051F4718911Cd0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A27180000051F4718911Cd0
  Replication:                                   none
  default_fencing:                               global

DID Device Name:                               /dev/did/rdsk/d13
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A27180000051B4718900Ad0
```

```
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A27180000051B4718900Ad0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A27180000051B4718900Ad0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A27180000051B4718900Ad0
  Replication:                                    none
  default_fencing:                                global

DID Device Name:                                /dev/did/rdsk/d14
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A2718000005174718 8D6Ed0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A27180000051747188D6Ed0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A27180000051747188D6Ed0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A27180000051747188D6Ed0
  Replication:                                    none
  default_fencing:                                global

DID Device Name:                                /dev/did/rdsk/d15
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A27180000051347187C26d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A27180000051347187C26d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A27180000051347187C26d0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A27180000051347187C26d0
  Replication:                                    none
  default_fencing:                                global

DID Device Name:                                /dev/did/rdsk/d16
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A27180000050F47187BE8d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A27180000050F47187BE8d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A27180000050F47187BE8d0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A27180000050F47187BE8d0
  Replication:                                    none
  default_fencing:                                global

DID Device Name:                                /dev/did/rdsk/d17
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A27180000050A47187BA0d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A27180000050A47187BA0d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A27180000050A47187BA0d0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A27180000050A47187BA0d0
  Replication:                                    none
  default_fencing:                                global

DID Device Name:                                /dev/did/rdsk/d18
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002AC8F2000005C8474EE2B1d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002AC8F2000005C8474EE2B1d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002AC8F2000005C8474EE2B1d0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002AC8F2000005C8474EE2B1d0
  Replication:                                    none
  default_fencing:                                global

DID Device Name:                                /dev/did/rdsk/d19
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A2F50000005E3474EE210d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A2F50000005E3474EE210d0
  Full Device Path:
```

```
lake9-mn:/dev/rdsk/c6t600A0B80002A2F50000005E3474EE210d0
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A2F50000005E3474EE210d0
  Replication:                                      none
  default_fencing:                                  nofencing

DID Device Name:                                /dev/did/rdsk/d20
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002AC8F2000005C6474EE28Fd0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002AC8F2000005C6474EE28Fd0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002AC8F2000005C6474EE28Fd0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002AC8F2000005C6474EE28Fd0
  Replication:                                      none
  default_fencing:                                  global

DID Device Name:                                /dev/did/rdsk/d21
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A2F50000005E1474EE1F0d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A2F50000005E1474EE1F0d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A2F50000005E1474EE1F0d0
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A2F50000005E1474EE1F0d0
  Replication:                                      none
  default_fencing:                                  nofencing

DID Device Name:                                /dev/did/rdsk/d22
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002AC8F2000005C4474EE26Fd0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002AC8F2000005C4474EE26Fd0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002AC8F2000005C4474EE26Fd0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002AC8F2000005C4474EE26Fd0
  Replication:                                      none
  default_fencing:                                  global

DID Device Name:                                /dev/did/rdsk/d23
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A2F50000005DF474EE1D0d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A2F50000005DF474EE1D0d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A2F50000005DF474EE1D0d0
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A2F50000005DF474EE1D0d0
  Replication:                                      none
  default_fencing:                                  nofencing

DID Device Name:                                /dev/did/rdsk/d24
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002AC8F2000005C2474EE251d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002AC8F2000005C2474EE251d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002AC8F2000005C2474EE251d0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002AC8F2000005C2474EE251d0
  Replication:                                      none
  default_fencing:                                  global

DID Device Name:                                /dev/did/rdsk/d25
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A2F50000005DD474EE1B0d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A2F50000005DD474EE1B0d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A2F50000005DD474EE1B0d0
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A2F50000005DD474EE1B0d0
```

```
  Replication:                                      none
  default_fencing:                                  nofencing

DID Device Name:                                 /dev/did/rdsk/d26
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002AC8F2000005C0474EE233d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002AC8F2000005C0474EE233d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002AC8F2000005C0474EE233d0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002AC8F2000005C0474EE233d0
  Replication:                                      none
  default_fencing:                                  global

DID Device Name:                                 /dev/did/rdsk/d27
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A2F50000005DB474EE198d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A2F50000005DB474EE198d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A2F50000005DB474EE198d0
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A2F50000005DB474EE198d0
  Replication:                                      none
  default_fencing:                                  nofencing

DID Device Name:                                 /dev/did/rdsk/d28
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A2F5000003C764947A1F6d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A2F5000003C764947A1F6d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A2F5000003C764947A1F6d0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A2F5000003C764947A1F6d0
  Replication:                                      none
  default_fencing:                                  global

DID Device Name:                                 /dev/did/rdsk/d29
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A2F50000005D9474EE17Ed0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A2F50000005D9474EE17Ed0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A2F50000005D9474EE17Ed0
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A2F50000005D9474EE17Ed0
  Replication:                                      none
  default_fencing:                                  nofencing

DID Device Name:                                 /dev/did/rdsk/d30
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002AC8F2000005BC474EE203d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002AC8F2000005BC474EE203d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002AC8F2000005BC474EE203d0
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002AC8F2000005BC474EE203d0
  Replication:                                      none
  default_fencing:                                  global

DID Device Name:                                 /dev/did/rdsk/d31
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A2F50000005D7474EDFA6d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A2F50000005D7474EDFA6d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A2F50000005D7474EDFA6d0
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A2F50000005D7474EDFA6d0
  Replication:                                      none
  default_fencing:                                  nofencing
```

```
DID Device Name:                                /dev/did/rdsk/d32
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002AC8F2000005B7474ED5D3d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002AC8F2000005B7474ED5D3d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002AC8F2000005B7474ED5D3d0
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002AC8F2000005B7474ED5D3d0
  Replication:                                     none
  default_fencing:                                 nofencing

DID Device Name:                                /dev/did/rdsk/d33
  Full Device Path:
lake8-mn:/dev/rdsk/c6t600A0B80002A2F50000005CF474ED4E6d0
  Full Device Path:
lake10-mn:/dev/rdsk/c3t600A0B80002A2F50000005CF474ED4E6d0
  Full Device Path:
lake9-mn:/dev/rdsk/c6t600A0B80002A2F50000005CF474ED4E6d0
  Full Device Path:
lake7-mn:/dev/rdsk/c6t600A0B80002A2F50000005CF474ED4E6d0
  Replication:                                     none
  default_fencing:                                 nofencing

DID Device Name:                                /dev/did/rdsk/d34
  Full Device Path:                                lake9-mn:/dev/rdsk/c0t0d0
  Replication:                                     none
  default_fencing:                                 global

DID Device Name:                                /dev/did/rdsk/d35
  Full Device Path:                                lake9-mn:/dev/rdsk/c0t1d0
  Replication:                                     none
  default_fencing:                                 global

DID Device Name:                                /dev/did/rdsk/d36
  Full Device Path:                                lake9-mn:/dev/rdsk/c1t0d0
  Replication:                                     none
  default_fencing:                                 global

DID Device Name:                                /dev/did/rdsk/d37
  Full Device Path:                                lake9-mn:/dev/rdsk/c3t0d0
  Replication:                                     none
  default_fencing:                                 global

DID Device Name:                                /dev/did/rdsk/d38
  Full Device Path:                                lake8-mn:/dev/rdsk/c0t0d0
  Replication:                                     none
  default_fencing:                                 global

DID Device Name:                                /dev/did/rdsk/d39
  Full Device Path:                                lake8-mn:/dev/rdsk/c0t1d0
  Replication:                                     none
  default_fencing:                                 global

DID Device Name:                                /dev/did/rdsk/d40
  Full Device Path:                                lake8-mn:/dev/rdsk/c1t0d0
  Replication:                                     none
  default_fencing:                                 global

DID Device Name:                                /dev/did/rdsk/d41
  Full Device Path:                                lake8-mn:/dev/rdsk/c3t0d0
  Replication:                                     none
  default_fencing:                                 global

DID Device Name:                                /dev/did/rdsk/d42
  Full Device Path:                                lake7-mn:/dev/rdsk/c0t0d0
  Replication:                                     none
  default_fencing:                                 global

DID Device Name:                                /dev/did/rdsk/d43
  Full Device Path:                                lake7-mn:/dev/rdsk/c0t1d0
  Replication:                                     none
  default_fencing:                                 global

DID Device Name:                                /dev/did/rdsk/d44
```

```
  Full Device Path:                                lake7-mn:/dev/rdsk/c1t0d0
  Replication:                                     none
  default_fencing:                                 global

DID Device Name:                                /dev/did/rdsk/d45
  Full Device Path:                                lake7-mn:/dev/rdsk/c3t0d0
  Replication:                                     none
```

```
default_fencing:                                        global
```

SAM-QFS Troubleshooting

Disaster Planning

- Disaster Planning and Recovery
- Backing Up Data
- Restoring Files and Directories
- Salvaging Damaged Volumes
- Recovering File Systems
- Recovering From Catastrophic Failure

SAM Troubleshooting

- Tools for SAM Troubleshooting
- Hardware Configuration Problems
- Troubleshooting the Configuration Files
- Troubleshooting the Archiver
- Troubleshooting the Releaser
- Troubleshooting the Recycler
- Using SAMreports

QFS Troubleshooting

- Recovering From a Hung mount Command
- Recovering From a Failed sammkfs Command
- Recovering From a Failed mount Command
- Checking File System Integrity and Repairing File Systems

SAM-QFS Manager Troubleshooting

- Troubleshooting SAM-QFS Manager

Checking File System Integrity and Repairing File Systems

Sun StorageTek QFS file systems write validation data in the following records that are critical to file system operations: directories, indirect blocks, and inodes. If the file system detects corruption while searching a directory, it issues an EDOM error, and the directory is not processed. If an indirect block is not valid, it issues an ENOCSI error, and the file is not processed. Table: Error Indicators summarizes these error indicators.

Error Indicators

Error	Solaris OS Meaning	Meaning
EDOM	Argument is out of domain.	Values in validation records are out of range.
ENOCSI	No CSI structure is available.	Links between structures are invalid.

In addition, inodes are validated and cross-checked with directories.

You should monitor the following files for error conditions:

- The log file specified in `/etc/syslog.conf` for the errors shown in Table: Error Indicators.
- The `/var/adm/messages` file for device errors.

If a discrepancy is noted, you should unmount the file system and check it using the `samfsck`(1M) command.

> **Note**
> The `samfsck`(1M) command can be issued on a mounted file system, but the results cannot be trusted. Because of this, you are encouraged to run the command on an unmounted file system only.

To Check a File System

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

```
samfsck -V <family-set-name>
```

  For family-set-name, specify the name of the file system as specified in the `mcf` file.

  You can send output from `samfsck`(1M) to both your screen and to a file by using it in conjunction with the `tee`(1) command, as follows.

  - C shell:

```
samfsck -V <family-set-name> |& tee <file>
```

  - Bourne shell:

```
samfsck -V <family-set-name> 2>&1 | tee <file>
```

Nonfatal errors returned by `samfsck`(1M) are preceded by `NOTICE`. Nonfatal errors are lost blocks and orphans. The file system is still consistent if `NOTICE` errors are returned. You can repair these nonfatal errors during a convenient, scheduled maintenance outage.

Fatal errors are preceded by ALERT. These errors include duplicate blocks, invalid directories, and invalid indirect blocks. The file system is not consistent if these errors occur. Notify Sun if the ALERT errors cannot be explained by a hardware malfunction.

If the `samfsck`(1M) command detects file system corruption and returns ALERT messages, you should determine the reason for the corruption. If hardware is faulty, repair it before repairing the file system.

For more information about the `samfsck`(1M) and `tee`(1) commands, see the `samfsck`(1M) and `tee`(1) man pages.

To Repair a File System

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

   Run the `samfsck`(1M) command when the file system is not mounted. For information about unmounting a file system, see ´appa§AHDDRª<olink targetptr="AHDDR">Unmounting a File System</olink>.
2. Use the `samfsck`(1M) command to repair a file system. If you are repairing a shared file system, issue the command from the metadata server.

   You can issue the `samfsck`(1M) command in the following format to repair a file system:

```
samfsck -F -V <fsname>
```

   For fsname, specify the name of the file system as specified in the `mcf` file.

Disaster Planning and Recovery

Contents

Disaster Planning and Recovery Guide

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

Many of the procedures in this section have to do with a Sun QFS file system running in conjunction with a Sun SAM storage and archive management system. This combination is sometimes referred to as SAM-QFS.

You, the system administrator, are assumed to be familiar with Sun QFS and Sun SAM installation, configuration, and basic operations. You are also assumed to be knowledgeable about the Solaris™ Operating System (Solaris OS) and network administration procedures, including installation, configuration, creation of accounts, and system backups.

For additional troubleshooting help, see SAM-QFS Troubleshooting.

Planning for Disaster Recovery

It is essential that you back up your data and establish disaster recovery processes so that data can be retrieved if any of the following occur:

- Data is accidentally deleted
- Storage media fail
- Systems fail

Broken Link (Target ID: CHAP04.SGM) provides the information you need to know about backing up metadata and other important configuration data. The remaining sections in this document describe how to use the data you back up to recover from various types of disasters.

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

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

Recovering From Failure of the Operating System Disk

When a disk containing the operating system fails, you must replace the defective disk and then perform bare metal recovery before you can do anything else. Two bare metal recovery approaches are available:

- Reinstalling the operating system, patches, and backed-up configuration files
  This process is slower than restoring a system image backup.
- Restoring a system image backup made ahead of time on a separate hard disk
  Image backups need to be made only when system configuration changes are made. The downside to this approach is that it is difficult to safely transport hard disks to off-site storage.

Testing Backup and Recovery Methods

After you have set up data recovery processes, you should perform the tests described in the following sections.

Testing Backup Scripts and `cron` Jobs

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

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

Testing the Disaster Recovery Process

Use the information in the other chapters in this manual to do the following tests in order to verify how well your disaster recovery process works. Do these tests periodically and any time you make changes to the software.

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

Preparing a Sun SAM or Sun QFS Disaster Recovery Plan

For SAM-QFS file systems, you should have the following in place in anticipation of needing them for disaster recovery:

- Up-to-date archive copies –
  The effectiveness of any of the SAM-QFS recovery methods relies primarily on frequent archiving. See Guidelines for Performing Metadata Dumps.
- Up-to-date metadata backups – See Metadata Used in Disaster Recovery.
- Archiver logs – If recent metadata is not available, archiver logs can help you re-create the file system directly from archive media. See Using Archiver Logs.

In addition, consider the following questions when preparing your site's disaster recovery plan:

- What is the right number of `samfsdump` or `qfsdump` files to retain at your site?
    - For a non-archiving file system, use the `qfsdump`(1M) command. This command generates a dump of both metadata and data. See [] for how to back up Sun SAM metadata.
    - For an archiving file system, use the `samfsdump`(1M) command. The `samfsdump` ~~u command dumps file data for files that do not have a current archive copy. The dump files are substantially larger with than without the~~ u option, and the command takes longer to complete. However, restoration of the output from `samfsdump -u` restores the file system back to its state when the dump was taken.
      The `samfsdump` command (without the - u option) generates a metadata dump file. A metadata dump file is relatively small, so you should be able to store many more metadata dump files than data dump files. Restoration of the output of `samfsdump` (without the ~~u option) is quicker than with the~~ u option, because the data is not restored until accessed by a user.
      Retain enough data and metadata to ensure that you can restore the file systems according to your site's needs. The appropriate number of dumps to save depends, in part, on how actively the system administrator monitors the dump output. If an administrator is monitoring the system daily to make sure the `samfsdump` or `qfsdump` dumps are succeeding and that there are enough tapes available, as well as investigating dump errors, then keeping a minimum number of dump files to cover vacations, long weekends, and other absences might be enough.
      If you are archiving data, are you actively recycling archive media? If so, be sure to schedule metadata copies to occur after recycling is complete.
      If your site is using the `sam-recycler`(1M) command to reclaim space on archive media, it is critical that you make metadata copies after `sam-recycler` has completed its work. If a metadata dump is created before `sam-recycler` exits, the information in the metadump about archive copies becomes out of date as soon as `sam-recycler` runs. Also, some archive copies may be made inaccessible because the `sam-recycler` command may cause archive media to be relabeled.
      Check `root`'s `crontab`(1) entry to find out if and when the `sam-recycler` command is being run, and then, if necessary, schedule the creation of metadump files around the `sam-recycler` execution times. For more about recycling, see Recycling.
- How much data should you store off site, and in what format?
  Off-site data storage is an essential part of a disaster recovery plan. In the event of a disaster, the only safe data repository might be an off-site vault. Beyond the recommended two copies of all files and metadata that you should be keeping in house as a safeguard against media failure, consider making a third copy on removable media and storing it off site.

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

- Is it sufficient to restore only the metadata to a pre-disaster state, or do you need also to restore all files that were online when the disaster happened?
    - The `samfsrestore`(1M) command can restore a file system to the state reflected in the `samfsdump` file. After the `samfsrestore`(1M) command is run, the metadata is restored, but the file data remains offline.
      If you need to restore all files that were online, you must run the `samfsrestore` command with the `-g` option.
      The log file generated by the `samfsrestore -g` command contains a list of all files that were on the disk when the `samfsdump`(1M) command was run. This log file can be used in conjunction with the `restore.sh` shell script to restore the files on disk to their pre-disaster state. The `restore.sh` script takes the log file as input and generates stage requests for files listed in the log. By default, the `restore.sh` script restores all files listed in the log file.
      If your site has thousands of files that need to be staged, consider splitting the log file into manageable chunks and running the `restore.sh` script against each of those chunks separately to ensure that the staging process does not overwhelm the system. You can also use this approach to ensure that the most critical files are restored first. For more information, see the comments in `/opt/SUNWsamfs/examples/restore.sh.`

> Note -
> If the `restore.sh` script is used in a shared environment, it must be run on the metadata server, not on one of the clients.

Next Steps

- Backing Up Data
- Restoring Files and Directories
- Salvaging Damaged Volumes
- Recovering File Systems
- Recovering From Catastrophic Failure

Hardware Configuration Problems

Sun StorageTek SAM problems can turn out to be hardware related. Before starting on an extensive troubleshooting exercise, determine the following:

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

To Verify Hardware

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

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

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

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

   The code example below shows the `st` tape devices checking in.

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

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

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

SAN-Attached Device Configuration Problems

Storage Area Network (SAN)-attached devices, such as Fibre Channel drives and automated libraries, should be checked to ensure that they are configured and that they are visible to the Solaris OS through the `cfgadm`(1M) command. Example: `cfgadm`(1M) Command Output illustrates this for a fabric-attached library controller and drives.

Example: ***cfgadm***(1M) Command Output

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

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

Recovering File Systems

Contents

Recovering File Systems

This chapter describes how to recover data when a SAM-QFS file system is corrupted or lost. The procedures vary, depending on the type of file system and whether you have recent `samfsdump`(1M) output for the file system. You might require assistance from your ASP or Sun Microsystems customer support.

Recovering a SAM-QFS File System With a Metadata Dump File

If you have `samfsdump`(1M) metadata output for a file system, you can use the `samfsrestore`(1M) command or SAM-QFS Manager to recover a file system that has been corrupted, accidentally remade, or destroyed. For details about the syntax and options used in the procedure, see the `samfsdump`(1M) man page or the SAM-QFS Manager online Help.

How to Restore a File System Using SAM-QFS Manager

1. From the Servers page, click the name of the server that contains the file system that you want.
   The File Systems Summary page is displayed.
2. Navigate to the File Browsing & Recovery node in the navigation tree.
   The File Browser page is displayed.
3. Use the file system mount point down menu to choose the file system you want to restore.
   A set of radio buttons displays on the top right corner of the table: Live Data and Recovery Point.

> Note -
> If you do not see these radio buttons, the selected recovery point has not been indexed. You must index the recovery point on the Recovery Points page before you can browse the recovery point file within the File Browser.

4. Select the Recovery Point radio button and choose a date of the recovery point you want to restore.
   The contents of the selected recovery point are displayed.
5. Locate the file you want to restore by browsing within the File Browser.
6. Select Restore Entire Recovery Point... from the Operations drop-down menu.
   The Restore window is displayed.
7. Specify the location to which you want to restore the file or directory.
   By default, the location is the path of the original file or directory, relative to the mount point of the file system. You can specify a different path relative to the mount point, or you can specify an absolute path on any archiving file system.
8. From the Online Status After Restoring drop-down menu, choose the actions that you want the file system to take after completing the restore process.
9. Click Submit.

How to Restore a File System Using the Command- Line Interface

1. Use the `cd`(1M) command to change to the mount point for the file system or to the directory in which you want to restore the file system.

> Caution -
> Consider restoring the file system first into a temporary directory and verifying that the restoration succeeds before restoring directly into the existing file system. This removes the risk of destroying the current file system before you can be sure the restoration is going to work. If the restoration fails, the file system may be recoverable by some other process.

In the following example, the mount point is `/sam1`.

```
# cd /sam1
```

1. Use the `samfsrestore` command with the `-T` and `-f` options to restore the entire file system relative to the current directory.
   Use the syntax shown in the following example, specifying the path name of the dump file after the `-f` option and the path name of the restore log file after the `-g` option.
   You can use the restore log file as input to the `restore.sh` script to stage back files that were online at the time of the dump.

```
# samfsrestore -T -f /dump_sam1/dumps/041126 -g _log_
```

Recovering a SAM-QFS File System Without a Dump File

You may be able to recover data from a SAM-QFS file system even if you do not have access to output from a samfsdump(1M) command, or to an archiver log file.

The following procedure shows you how to re-create user files by reloading tape or optical disk and using the star(1M) command's{{– n}} option.

> Note -
> Recovering file systems from archive cartridges and using the star command is a tedious and time-consuming process. This should not be considered the standard procedure for disaster recovery.

How to Recover a File System Without a Dump File

1. (Optional) Disable any automated processes that are related to Sun StorageTek SAM operations.
   If any of the following automated processes are running, disable them during the recovery process to ensure that no data is lost:
   - Recycling – Disable any recycling activities, including those triggered by an entry in root's crontab(4). Failure to disable recycling activity could result the recycling and relabeling of tapes that contain active data.
   - Archiving
   - Processes that capture samfsdump(1M) files – Suspending these processes saves an existing samfsdump output file, and provides an opportunity for easier recovery.
   - Writes into the file system
2. (Optional) Disable network file system (NFS) sharing for the file system.
   It can be easier to recover data if the file system is not running NFS sharing on the file systems during the recovery period.
3. Use the sammkfs(1M) command to remake the SAM-QFS file system to be restored.
4. Identify the cartridges that contain the archive copy information.
5. Read all the archive media.
   If you are using tapes, use the tar(1M), gnutar(1M), or star(1M) command.
6. If recovering from tape media, use the tarback.sh script.
   The tarback.sh script is located in /opt/SUNWsamfs/examples/tarback.sh. This script identifies a single tape drive for use during recovery and provides a list of VSNs to recover. The script uses star(1M) to loop through a volume, reading all available archive files.
   The star(1M) command is an enhanced version of gnutar(1M). The tarback.sh script uses star(1M) and the -n option, which restores only files that are newer than the existing copy. If the archive copy that you are about to restore is older than the existing copy, the restore is skipped.
   The tarback.sh script is described in Backup and Recovery Commands and Tools. For more information about this script, see the tarback.sh man page. See also Unreadable Label With No Other Copies Available for an example of how to use the script.
7. If recovering from magnetic-optical media, contact Sun customer support.

Recovering From a Failed mount Command

A mount(1M) command can fail for several reasons. This section describes some actions you can take to remedy a mount problem. If the mount(1M) command hangs, rather than fails, see Recovering From a Hung mount(1M) Command.

Some failed mount(1M) behaviors and their remedies are as follows:

- If the mount(1M) command fails with a Shared server is not mounted message generated on a client, determine the server host and mount the file system on the metadata server.
- If the mount command fails with a message indicating that there is a mismatch between the file system and the mcf file, ensure the following:
  - The mcf file is syntactically valid. For more information, see To Verify the mcf File and Propagate mcf File Changes to the System.
  - Recent changes to the mcf file are valid and have been enacted. For more information, see To Verify the mcf File and Propagate mcf File Changes to the System.
  - The mcf file matches the server's mcf file with device names or controller numbers adjusted for any differences on the client.

You can use the samfsconfig(1M) command to diagnose some of these problems. For more information about using the samfsconfig(1M) command, see To Use the samfsconfig(1M) Command.

If the mount(1M) command fails for other reasons, use the following procedures to verify the system characteristics that must be in place in order for the mount(1M) command to be successful.

- To Verify that the File System Can Be Mounted
- To Use the samfsinfo(1M) and samsharefs(1M) Commands
- To Use the samfsconfig(1M) Command

To Verify that the File System Can Be Mounted

If this procedure does not expose errors, perform To Use the samfsinfo(1M) and samsharefs(1M) Commands, which can help you verify that the file system has been created and that the shared hosts file is correctly initialized.

The following procedure shows you what to verify if the mount(1M) command fails.

1. Ensure that the mount point directory is present.

   There are multiple ways to accomplish this. For example, you can issue the ls(1) command in the following format:

   ```
   ls -ld <mountpoint>
   ```

   For mountpoint, specify the name of the Sun StorageTek QFS shared file system's mount point.

   When you examine the ls(1) command's output, make sure that the output shows a directory with access mode 755. In other words, the codes should read drwxr-xr-x. The following code shows example output.

   ```
   # ls -ld /sharefs1
   drwxr-xr-x   2 root     sys          512 Mar 19 10:46 /sharefs1
   ```

   If the access is not at this level, enter the following chmod(1) command:

   ```
   # chmod 755 <mountpoint>
   ```

   For mountpoint, specify the name of the Sun StorageTek QFS shared file system's mount point.

2. Ensure that there is an entry for the file system in the /etc/vfstab file.

   The following code shows an entry for the shared file system named sharefs1.

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

   Ensure that the shared flag is present in the Mount Parameters field of the shared file system's entry in the /etc/vfstab file.

3. Ensure that the mount point directory is not shared out for NFS use.

   If the mount point is shared, use the unshare(1M) command to unshare it. For example:

   ```
   # unshare mountpoint
   ```

   For mountpoint, specify the name of the Sun StorageTek QFS shared file system's mount point.

To Use the **samfsinfo**(1M) and **samsharefs**(1M) Commands

This procedure shows how to analyze the output from these commands.

1. Enter the `samfsinfo`(1M) command on the server.

   Use this command in the following format:

   ```
   samfsinfo <filesystem>
   ```

   For filesystem, specify the name of the Sun StorageTek QFS shared file system as specified in the `mcf` file. The following code shows the `samfsinfo`(1M) command and output.

   ```
   titan-server# samfsinfo sharefs1
   samfsinfo: filesystem sharefs1 is mounted.
   name:     sharefs1       version:     2    shared
   time:     Mon Apr 29 15:12:18 2002
   count:    3
   capacity:      10d84000          DAU:         64
   space:         10180400
   meta capacity: 009fe200          meta DAU:    16
   meta space:    009f6c60
   ord  eq   capacity      space   device
   1    11   086c0000   080c39b0   /dev/dsk/c1t2100002037E9C296d0s6
   2    12   086c4000   080bca50   /dev/dsk/c3t50020F2300005D22d0s6
   3    13   086c4000   080a9650   /dev/dsk/c3t50020F2300006099d0s6
   4    14   086c4000   08600000   /dev/dsk/c3t50020F230000651Cd0s6
   ```

   The output from the code above shows a `shared` keyword in the following line:

   ```
   name:     sharefs1       version:     2    shared
   ```

   Note the list of file system devices, ordinalsordinals, and equipment numbers that appear after the following line:

   ```
   ord  eq   capacity      space   device
   ```

   Make sure that these numbers correspond to the devices in the file system's `mcf`(4) entry.

2. Enter the `samsharefs`(1M) command on the server.

   Use this command in the following format:

   ```
   samsharefs -R <filesystem>
   ```

   For filesystem, specify the name of the Sun StorageTek QFS shared file system as specified in the `mcf` file. The following code example shows the `samsharefs`(1M) command and output.

   ```
   titan-server# samsharefs -R sharefs1
   #
   # Host file for family set `sharefs1'
   #
   # Version: 3    Generation: 50    Count: 4
   # Server = host 0/titan, length = 216
   #
   titan 173.26.2.129,titan.foo.com 1 - server
   tethys 173.26.2.130,tethys.foo.com 2 -
   dione dione.foo.com 0 -
   mimas mimas.foo.com 0 -
   ```

   The following information pertains to the diagnostic output from the `samfsinfo`(1M) or `samsharefs`(1M) commands.

- If either command issues diagnostics or error messages, resolve them. Ensure that the output from the samfsinfo(1M) command includes the shared keyword.
- You can execute these commands on alternate server hosts and on client hosts that have no nodev devices in the host's mcf (4) entry for the file system.

  If the samfsinfo (1M) and samsharefs(1M) commands do not expose irregularities, perform To Use the samfsconfig(1M) Command.

To Use the **samfsconfig**(1M) Command

On clients with nodev device entries in the mcf file for the file system, the entire file system might not be accessible, and the shared hosts file might not be directly accessible. You can use the samfsconfig(1M) command to determine whether the shared file system's data partitions are accessible.

1. Issue the samfsconfig(1M) command.

   Use this command in the following format:

```
samfsconfig <list-of-devices>
```

   For list-of-devices, specify the list of devices from the file system entry in the mcf file. Use a space to separate multiple devices in the list.

   Example 1. The following code example shows the mcf file for the host tethys, a host that does not have a nodev entry in its mcf file. It then shows the samfsconfig(1M) command issued.

```
tethys# cat /etc/opt/SUNWsamfs/mcf
sharefs1                         10  ma   sharefs1    on  shared
/dev/dsk/c1t2100002037E9C296d0s6 11  mm   sharefs1    -
/dev/dsk/c3t50020F2300005D22d0s6 12  mr   sharefs1    -
/dev/dsk/c3t50020F2300006099d0s6 13  mr   sharefs1    -
/dev/dsk/c3t50020F230000651Cd0s6 14  mr   sharefs1    -
tethys# samfsconfig /dev/dsk/c1t2100002037E9C296d0s6 /dev/dsk/c3t50020F2300005D22d0s6
/dev/dsk/c3t50020F2300006099d0s6 /dev/dsk/c3t50020F230000651Cd0s6
#
# Family Set `sharefs1' Created Mon Apr 29 15:12:18 2002
#
sharefs1                           10    ma   sharefs1  - shared
/dev/dsk/c1t2100002037E9C296d0s6   11    mm   sharefs1  -
/dev/dsk/c3t50020F2300005D22d0s6   12    mr   sharefs1  -
/dev/dsk/c3t50020F2300006099d0s6   13    mr   sharefs1  -
/dev/dsk/c3t50020F230000651Cd0s6   14    mr   sharefs1  -
```

   Example 2. The following code example shows the samfsconfig(1M) command being used on a host that has a nodev entry in its mcf file.

```
dione# cat /etc/opt/SUNWsamfs/mcf
sharefs1                             10    ma   sharefs1  on  shared
nodev                              11    mm   sharefs1  -
/dev/dsk/c4t50020F23000055A8d0s3   12    mr   sharefs1  -
/dev/dsk/c4t50020F23000055A8d0s4   13    mr   sharefs1  -
/dev/dsk/c4t50020F23000055A8d0s5   14    mr   sharefs1  -
dione# samfsconfig /dev/dsk/c4t50020F23000055A8d0s3 /dev/dsk/c4t50020F23000055A8d0s4
/dev/dsk/c4t50020F23000055A8d0s5
# Family Set `sharefs1' Created Mon Apr 29 15:12:18 2002
# Missing slices
# Ordinal 1
# /dev/dsk/c4t50020F23000055A8d0s3    12    mr   sharefs1  -
# /dev/dsk/c4t50020F23000055A8d0s4    13    mr   sharefs1  -
# /dev/dsk/c4t50020F23000055A8d0s5    14    mr   sharefs1  -
```

   For examples 1 and 2, verify that the output lists all slices from the file system, other than the metadata (mm) devices, as belonging to the file system. This is the case for example 2.

Recovering From a Failed samd config Command

Recovering From a Failed **samd config** Command

Symptom

When you run the `samd config` command for a file system, you might see an error message.

What Happened

The `sam-fsd` daemon or some other service upon which SAM-QFS relies is not running.

What to Do

To resolve this problem, determine whether the `sam-fsd` daemon or some other service is at fault.
Use the following form of the `svcs`(1) command:

```
# svcs -xv
```

This command shows all services that are defined, but are in a maintenance state.

You can also use SMF to determine the specific state of a SAM-QFS service. For example:

```
# svcs sam-fsd
STATE          STIME    FMRI
online         10:17:38 svc:/system/sam-fsd:default
#
```

More Information

The Solaris Management Facility (SMF) manages the `sam-fsd` daemon. When you install the Sun QFS or Sun Storage Archive Manager packages on a Solaris host, the `sam-fsd` SMF service is installed, but is disabled by default. The `sam-fsd` service is enabled automatically upon use of SAM-QFS, such as a `samd config` command.

SMF can manage error conditions that are unique to SAM-QFS during its initialization, such as Sun QFS file system automount and SAM-QFS configuration. This management ability enables normal system operations to continue. SMF management includes such tasks as stopping or restarting SAM-QFS daemons as needed, and ignoring some QFS file system and archiving misconfigurations. Error messages are written to the appropriate SMF services logs.

For more information about SMF, see the `smf`(5) man page.

Recovering From a Failed sammkfs Command

The procedures in this section can be performed on client hosts and can also be performed on the server. Commands that can be executed only on the metadata server are preceded with a `server#` prompt.

If the `sammkfs`(1M) command returns an error or messages indicating that an unexpected set of devices are to be initialized, you need to perform this procedure. It includes steps for verifying the `mcf` file and for propagating `mcf` file changes to the system.

To Verify the **mcf** File and Propagate **mcf** File Changes to the System

1. Use the `sam-fsd`(1M) command to verify the `mcf` file.

   For example:

   ```
   # sam-fsd
   ```

Examine the output from the sam-fsd(1M) command and determine if there are errors that you need to fix.

2. If the output from the sam-fsd(1M) command indicates that there are errors in the /etc/opt/SUNWsamfs/mcf file, edit the mcf file to resolve these issues.
3. Issue the sam-fsd(1M) command again to verify the mcf file.

   Repeat Step 1, Step 2, and Step 3 of this process until the output from the sam-fsd(1M) command indicates that the mcf file is correct.
4. Issue the samd(1M) config command.

   This is needed to propagate mcf file changes by informing the sam-fsd daemon of the configuration change.

   For example:

```
# samd config
```

Recovering From a Hung mount Command

If the mount(1M) command hangs, follow the procedure in this section. You have a hung mount(1M) command if, for example, the mount(1M) command fails with a connection error or with a Server not responding message that does not resolve itself within 30 seconds.

The most typical remedy for a hung mount(1M) command is presented first. If that does not work, perform the subsequent procedures.

To Verify Network Connections

The netstat(1M) command verifies that the sam-sharefsd daemon's network connections are correctly configured.

1. Become superuser on the metadata server.
2. Type the samu(1M) command to invoke the samu(1M) operator utility.

   For example:

```
# samu
```

3. Press :P to access the Active Services display.

   The following code example shows a P display.

```
Active Services                        samu   4.4 09:02:22 Sept 22 2005
Registered services for host `titan':
    sharedfs.sharefs1
  1 service registered.
```

   Examine the output. In the above code example, look for a line that contains sharedfs.filesystem-name. In this example, the line must contain sharedfs.sharefs1.

   If no such line appears, you need to verify that both the sam-fsd and sam-sharefsd daemons have started. Perform the following steps:

   a. Enable daemon tracing in the defaults.conf file.

      For information about how to enable tracing, see defaults.conf(4) or see Step 2 in To Examine the sam-sharefsd Trace Log .
   b. Examine your configuration files, especially /etc/opt/SUNWsamfs/mcf.
   c. After you have checked your configuration files and verified that the daemons are active, begin this procedure again.
4. Enter the samsharefs(1M) command to check the hosts file.

   The following code example shows the samsharefs(1M) command and correct output.

```
titan-server# samsharefs -R sharefs1
#
# Host file for family set `sharefs1'
#
# Version: 3    Generation: 50    Count: 4
# Server = host 0/titan, length = 216
#
titan 173.26.2.129 1 - server
tethys 173.26.2.130 2 -
dione dione 0 -
mimas mimas 0 -
```

In the output on your system, verify the following:

- The host name is present in column 1 of the output and it is designated as the server.
- The host IP address is present in column 2. If there are multiple IP addresses, make sure that they are all valid.

5. Enter the `netstat`(1M) command on the server.

   The following code example shows the `netstat`(1M) command entered on server `titan`.

```
titan-server# netstat -a | grep sam-qfs
      *.sam-qfs *.*            0     0 24576  0 LISTEN
      *.sam-qfs *.*            0     0 24576  0 LISTEN
titan.32834  titan.sam-qfs 32768     0 32768  0 ESTABLISHED
titan.sam-qfs  titan.32891 32768     0 32768  0 ESTABLISHED
titan.sam-qfs tethys.32884 24820     0 24820  0 ESTABLISHED
titan.sam-qfs  dione.35299 24820     0 24820  0 ESTABLISHED
     *.sam-qfs *.*             0     0 24576  0 LISTEN
```

   Verify that the output from the `netstat`(1M) command on the server contains the following:

   - Three `LISTEN` entries.
   - Two `ESTABLISHED` entries for the host.
   - One `ESTABLISHED` entry for each client that is configured and running the `sam-fsd` daemon.
     This example shows `ESTABLISHED` entries for `tethys` and `dione`. There should be one `ESTABLISHED` entry for each client that is configured and running, whether or not it is mounted.

6. Enter the `netstat`(1M) command on the client.

   The following code example shows the `netstat`(1M) command entered on client `dione`.

```
titan-server# netstat -a | grep sam-qfs
      *.sam-qfs *.*            0     0 24576  0 LISTEN
      *.sam-qfs *.*            0     0 24576  0 LISTEN
titan.32834  titan.sam-qfs 32768     0 32768  0 ESTABLISHED
titan.sam-qfs  titan.32891 32768     0 32768  0 ESTABLISHED
titan.sam-qfs tethys.32884 24820     0 24820  0 ESTABLISHED
titan.sam-qfs  dione.35299 24820     0 24820  0 ESTABLISHED
     *.sam-qfs *.*             0     0 24576  0 LISTEN
```

7. Verify that the output contains the following:
   - Three `LISTEN` entries. All entries are for the `sam-fsd` daemon.
   - One `ESTABLISHED` entry.

     If these lines are present, then the network connection is established.

     If an `ESTABLISHED` connection is not reported, perform one or more of the following procedures:

- To Verify That the Client Can Reach the Server
- To Verify That the Server Can Reach the Client
- To Examine the sam-sharefsd Trace Log

To Verify That the Client Can Reach the Server

Perform these steps if using the procedure described in To Verify Network Connections did not show an ESTABLISHED connection.

1. Use the samsharefs(1M) command to verify the hosts file on the server.

   You can issue the samsharefs(1M) command on alternate server hosts and client hosts that have no nodev devices listed in the host's mcf(4) entry for the file system. For this step, use this command in the following format:

```
samsharefs -R <filesystem>
```

   For filesystem, specify the name of the Sun StorageTek QFS shared file system as specified in the mcf file. The following code example shows the samsharefs(1M) -R command.

```
titan-server# samsharefs -R sharefs1
#
# Host file for family set `sharefs1'
#
# Version: 3    Generation: 50    Count: 4
# Server = host 0/titan, length = 216
#
titan 173.26.2.129 1 - server
tethys 173.26.2.130 2 -
dione dione 0 -
mimas mimas 0 -
```

2. Save this output.

   If the steps in this procedure fail, you need this output for use in subsequent procedures.
3. Verify that the output matches expectations.

   If the command fails, verify that the file system was created. In this case it is likely that one of the following has occurred:

- The mcf file was not created properly. You can use the samfsconfig(1M) command to verify the correctness of the mcf file.
- The file system was never created.
- The initial hosts configuration files have not been created. The configuration process involves editing the existing mcf file, propagating the mcf file changes to the rest of the system, and configuring the hosts files.

1. Find the row containing the server's name in the first column.
2. From the client, use the ping(1M) command on each entry from the second column of samsharefs(1M) output to verify that the server can be reached.

   Use this command in the following format:

```
ping <servername>
```

   For servername, specify the name of the server as shown in the second column of the samsharefs(1M) command's output.

   The following code example shows output from ping(1M).

```
dione-client# ping 173.26.2.129
ICMP Host Unreachable from gateway dione (131.116.7.218)
for icmp from dione (131.116.7.218) to 173.26.2.129
dione-client# ping titan
titan.foo.com is alive
```

3. If the ping(1M) command revealed unreachable hosts, examine the hosts.}}filesystem{{.local file from the client.

   If there is more than one entry in the second column of samsharefs(1M) output, and if some of the entries are not reachable, ensure that only the reachable entries for the entries you want the shared file system to use are present. Also ensure that the necessary entries are present in the /etc/opt/SUNWsamfs/hosts.filesystem.local file entry on that host. Ensure that the unreachable hosts are not entered in these places.

If the sam-sharefsd daemon attempts to connect to unreachable server interfaces, there can be substantial delays in its connecting to the server after installation, rebooting, or file system host reconfiguration. This affects metadata server failover operations substantially.

The following code example shows the hosts.sharefs1.local file.

```
dione-client# cat /etc/opt/SUNWsamfs/hosts.sharefs1.local
titan       titan # no route to 173.26.2.129
tethys      tethys # no route to 173.26.2.130
```

4. If the ping(1M) command revealed that there were no reachable server interfaces, enable the correct server interfaces.

   Either configure or initialize the server network interfaces for typical operations, or use the samsharefs(1M) command to update the interface names in the hosts file so they match the actual names.

To Verify That the Server Can Reach the Client

Perform these steps if the procedure in To Verify Network Connections did not show an ESTABLISHED connection.

1. Obtain samsharefs(1M) output.

   This can be the output generated in To Verify That the Client Can Reach the Server, or you can generate it again using the initial steps in that procedure.
2. Find the row containing the client's name in the first column.
3. On the client, run the hostname(1M) command and ensure that the output matches the name in the first column of samsharefs (1M) output.

   The following code example shows the hostname(1M) command and its output.

```
dione-client# hostname
dione
```

4. If the hostname(1M) command output matched the name in the second column of samsharefs(1M) output, use the ping(1M) command on the server to verify that the client can be reached.

   The following code example shows the ping(1M) command and its output.

```
titan-server# ping dione
dione is alive
```

   It is not necessary that every entry in column two of the code example in Step 6 be reachable, but all interfaces that you wish any potential server to accept connections from must be present in the column. The server rejects connections from interfaces that are not declared in the shared hosts file.

5. If the ping(1M) command revealed that there were no reachable client interfaces, enable the correct client interfaces.

   Either configure or initialize the client network interfaces for typical operations, or use the samsharefs(1M) command to update the interface names in the hosts file so they match the actual names.

To Examine the **sam-sharefsd** Trace Log

The trace log files keep information generated by the sam-sharefsd(1M) daemons during their operation. The trace log files include information about connections attempted, received, denied, refused, and so on, as well as other operations such as host file changes and metadata server changes.

Tracking problems in log files often involves reconciling the order of operations on different hosts by using the log files. If the hosts' clocks are synchronized, log file interpretation is greatly simplified. One of the installation steps directs you to enable the network time daemon, xntpd (1M). This synchronizes the clocks of the metadata server and all client hosts during Sun StorageTek QFS shared file system operations.

The trace logs are particularly useful when setting up an initial configuration. The client logs show outgoing connection attempts. The corresponding messages in the server log files are some of the most useful tools for diagnosing network and configuration problems with the

Sun StorageTek QFS shared file system. The log files contain diagnostic information for resolving most common problems.

The following procedures can resolve most mount(1M) problems:

- To Verify Network Connections</olink>
- To Verify That the Client Can Reach the Server
- To Verify That the Server Can Reach the Client

If none of the preceding procedures resolve the problem, perform the steps in this section. You can perform these steps on both the server and the client hosts.

1. Verify the presence of file /var/opt/SUNWsamfs/trace/sam-sharefsd.

   If this file is not present, or if it shows no recent modifications, proceed to the next step.

   If the file is present, use tail(1) or another command to examine the last few lines in the file. If it shows suspicious conditions, use one or more of the other procedures in this section to investigate the problem.
2. If Step 1 indicates that file /var/opt/SUNWsamfs/trace/sam-sharefsd does not exist or if the file shows no recent modifications, edit file /etc/opt/SUNWsamfs/defaults.conf and add lines to enable sam-sharefsd tracing.
   a. If a defaults.conf file does not already reside in /etc/opt/SUNWsamfs, copy the example defaults.conf file from /opt/SUNWsamfs/examples/defaults.conf to /etc/opt/SUNWsamfs:

```
# cd /etc/opt/SUNWsamfs
# cp /opt/SUNWsamfs/examples/defaults.conf .
```

   b. Use vi(1) or another editor to edit file /etc/opt/SUNWsamfs/defaults.conf and add lines to enable tracing.

      The following code example shows the lines to add to the defaults.conf file.

```
trace
sam-sharefsd = on
sam-sharefsd.options = all
endtrace
```

   c. Issue the samd(1M) config command to reconfigure the sam-fsd(1M) daemon and cause it to recognize the new defaults.conf file.

      For example:

```
# samd config
```

   d. Issue the sam-fsd(1M) command to check the configuration files.

      The following code example shows the output from the sam-fsd(1M) command.

```
# sam-fsd
Trace file controls:
sam-archiverd off
sam-catserverd off
sam-fsd       off
sam-rftd      off
sam-recycler  off
sam-sharefsd  /var/opt/SUNWsamfs/trace/sam-sharefsd
              cust err fatal misc proc date
              size    0    age 0
sam-stagerd   off
Would stop sam-archiverd()
Would stop sam-rftd()
Would stop sam-stagealld()
Would stop sam-stagerd()
Would stop sam-initd()
```

e. Examine the log file in /var/opt/SUNWsamfs/trace/sam-sharefsd to check for errors:

```
# more /var/opt/SUNWsamfs/trace/sam-sharefsd
```

3. Examine the last few dozen lines of the trace file for diagnostic information.

   The following code example shows a typical sam-sharefsd client log file. In this example, the server is titan, and the client is dione. This file contains normal log entries generated after a package installation, and it finishes with the daemon operating normally on a mounted file system.

```
dione# tail -18 /var/opt/SUNWsamfs/trace/sam-sharefsd
2004-03-23 16:13:11 shf-shsam2[13835:1]: FS shsam2: Shared file system daemon started -
config only
2004-03-23 16:13:11 shf-shsam2[13835:1]: FS shsam2: Host dione
2004-03-23 16:13:11 shf-shsam2[13835:1]: FS shsam2: Filesystem isn't mounted
2004-03-23 16:13:11 shf-shsam2[13837:1]: FS shsam2: Shared file system daemon started
2004-03-23 16:13:11 shf-shsam2[13837:1]: FS shsam2: Host dione
2004-03-23 16:13:11 shf-shsam2[13837:1]: FS shsam2: Filesystem isn't mounted
2004-03-23 16:13:11 shf-shsam2[13837:1]: FS shsam2: Kill sam-sharefsd pid 13835
2004-03-23 16:13:12 shf-shsam2[13837:1]: FS shsam2: Killed sam-sharefsd pid 13835
2004-03-23 16:13:12 shf-shsam2[13837:1]: FS shsam2: Host dione; server = titan
2004-03-23 16:13:12 shf-shsam2[13837:1]: FS shsam2: Wakened from AWAIT_WAKEUP
2004-03-23 16:13:14 shf-shsam2[13837:5]: FS shsam2: Set Client (Server titan/3).
2004-03-23 16:13:14 shf-shsam2[13837:5]: FS shsam2: SetClientSocket dione (flags=0)
2004-03-23 16:13:14 shf-shsam2[13837:5]: FS shsam2: rdsock dione/0 (buf=6c000).
2004-03-23 16:13:15 shf-shsam2[13837:1]: FS shsam2: Signal 1 received: Hangup
2004-03-23 16:13:15 shf-shsam2[13837:1]: FS shsam2: Wakened from AWAIT_WAKEUP
2004-03-23 16:13:15 shf-shsam2[13837:1]: FS shsam2: mount; flags=18889
2004-03-23 16:18:55 shf-shsam2[13837:1]: FS shsam2: Signal 1 received: Hangup
2004-03-23 16:18:55 shf-shsam2[13837:1]: FS shsam2: Wakened from AWAIT_WAKEUP
```

Recovering From Catastrophic Failure

Contents

Recovering From Catastrophic Failure

Certain events, such as flooding in a computer room, can be classified as catastrophic failures. This section describes the procedure to follow after such an event. You might require the assistance of your ASP or Sun Microsystems customer support.

Recovery Task Overview

You should not recover any system component, software element, or file system that has not failed. However, you might need to reconfigure the file system on a restored system to regain access to file systems or to determine whether any file system has failed.

The process of recovering from a catastrophic failure involves the following tasks:

Task	For More Information
Determine the failed system component.	How to Restore Failed System Components

Disable the archiver and the recycler until all files are restored.	How to Disable the Archiver and Recycler Until All Files Are Restored
Compare previous and current configuration files, and reconcile inconsistencies.	How to Keep and Compare Previous and Current Configuration and Log Files
Repair disks.	How to Repair Disks
Restore or build new library catalog files.	How to Restore or Build New Library Catalog Files
Create new file systems and restore from `samfsdump` output.	How to Make New File Systems and Restore From `samfsdump` Output

How to Restore Failed System Components

1. Ascertain which components have failed.
2. If a hardware component has failed, restore it to operation, preserving any available data.
   If the failing component is a disk drive that has not totally failed, preserve as much information as possible. Before replacing or reformatting the disk, identify any salvageable files, and copy these files to a tape or to another disk for future use in the recovery process. Salvageable files to identify and copy include the following:
   - File system dumps
   - Sun SAM configuration files, archiver log files, or library catalogs
3. If the Solaris Operating System (OS) has failed, restore it to operation.
   See Recovering From Failure of the Operating Environment Disk. Verify that the Solaris OS is functioning correctly before proceeding.
4. If the Sun SAM or Sun QFS package has been damaged, remove and reinstall it from a backup copy or from its distribution file.
   You can verify whether a package has been damaged by using the `pkgchk`(1M) utility.
5. If disk hardware used by the Sun SAM software was repaired or replaced in Step 2, configure the disks (for RAID binding or mirroring) as necessary.
   Reformat disks only if they have been replaced or if it is otherwise absolutely necessary.

How to Disable the Archiver and Recycler Until All Files Are Restored

Before You Begin

If the recycler is enabled so that it runs before all files are restored, cartridges with good archive copies might be improperly relabeled.

Steps

1. Add a single global `wait` directive to the `archiver.cmd` file or add a file-system-specific wait directive for each file system for which you want to disable archiving.
2. Open the `/etc/opt/SUNWsamfs/archiver.cmd` file for editing and find the section in which you want to insert the `wait` directive.
   In the following sample file, local archiving directives exist for two file systems, `samfs1` and `samfs2`.

```
# vi /etc/opt/SUNWsamfs/archiver.cmd
...
fs = samfs1
allfiles   .
1   10s
fs = samfs2
allfiles   .
1   10s
```

3. Add the `wait` directive.
   - To apply the directive globally, insert it before the first `fs =` command (`fs = samfs1`), as shown here:

```
wait
fs = samfs1
allfiles   .
1   10s
fs = samfs2
allfiles   .
1   10s
:wq
```

- To apply the directive to a single file system, insert it after the `fs =` command for that file system, as shown here:

```
fs = samfs1
wait
allfiles   .
1   10s
fs = samfs2
wait
allfiles   .
1   10s
:wq
```

4. Add a global `ignore` directive to the `recycler.cmd` file, or add a file-system-specific `ignore` directive for each library for which you want to disable recycling.
5. Open the `/etc/opt/SUNWsamfs/recycler.cmd` file for editing, as shown in the following example.

```
# _vi /etc/opt/SUNWsamfs/recycler.cmd_
...
         logfile = /var/adm/recycler.log
         lt20 -hwm 75 -mingain 60
         lt20 75 60
         hp30 -hwm 90 -mingain 60 -mail root
         gr47 -hwm 95 -mingain 60 -mail root
```

6. Add the `ignore` directives.
   The following example shows `ignore` directives added for three libraries.

```
#  recycler.cmd.after  - example recycler.cmd file
#
         logfile = /var/adm/recycler.log
         lt20 -hwm 75 -mingain 60 -ignore
         hp30 -hwm 90 -mingain 60 -ignore -mail root
         gr47 -hwm 95 -mingain 60 -ignore -mail root
```

How to Keep and Compare Previous and Current Configuration and Log Files

Follow these steps before rebuilding the system.

1. Recover any available Sun SAM configuration files or archiver log files from the system's disks.
2. Compare the restored versions of all configuration files represented in the }}{{`SAMreport` with those restored from the system backups.
3. If inconsistencies exist, determine the effect of the inconsistencies and reinstall the file system, if necessary, using the configuration information in the `SAMreport` file.
   For more information on `SAMreport` file, see the `samexplorer`(1M) man page.

How to Repair Disks

1. For file systems that reside on disks that have not been replaced, run the `samfsck`(1M) utility to repair small inconsistencies, reclaim

lost blocks, and so on.
For command-line options to the samfsck utility, see the samfsck(1M) man page.

How to Restore or Build New Library Catalog Files

1. Replace the most recent library catalog file copies from the removable media files, from the Sun StorageTek SAM server disks, or from the most recent file system archive copies.
2. If the library catalogs are unavailable, build new catalogs by using the build.cat(1M) command and the library catalog section of the most recent SAMreport as input.
   Use the newest library catalog copy available for each automated library.

   > Note -
   > Sun SAM systems automatically rebuild library catalogs for SCSI-attached automated libraries. This does not occur for ACSLS-attached automated libraries. Tape usage statistics are lost.

How to Make New File Systems and Restore From **samfsdump** Output

Follow these steps for file systems that were partially or completely resident on disks that were replaced or reformatted.

1. Obtain the most recent copy of the samfsdump(1M) output file.
2. Make a new file system and restore the file system using the samfsdump output file.
3. Use the sammkfs (1M) command to make a new file system.

```
# mkdir /sam1
# sammkfs samfs1
# mount samfs1
```

4. Use the samfsrestore (1M) command with the -f option and the -g option, use the following syntax:

```
samfsrestore -f <output-file-location> -g <log-file>
```

   where:

   - output-file-location is the location of the samfsdump output file.
   - log-file is the path name of the new log file that will list all the files that were online.
     For example:

```
# cd /sam1
# samfsrestore -f /dump_sam1/dumps/040120 -g /var/adm/messages/restore_log
```

   > Note -
   > Once all file systems have been restored, the system can be made available to users in degraded mode.

5. On the file systems you have just restored, perform the following steps:
   a. Run the restore.sh script against the log file, and stage all files that were known to be online before the outage. In a shared environment, this script must be run on the metadata server.
   b. Run the sfind (1M) command against the file system to determine which files are labeled as damaged.
      These files might or might not be restorable from tape, depending on the content of the archive log files. Determine the most recently available archive log files from one of the following sources, in this order:
      - The removable media file.
      - The Sun SAM server disk.
      - The most recent file system archive. This source is likely to be slightly outdated.
   c. Run the grep(1) command against the most recent archive log file to search for the damaged files.
      This will enable you to determine whether any of the damaged files were archived to tape after the last time the samfsdump

(1M) command was run.

   d. Examine the archive log files to identify any archived files that do not exist in the file system.

6. Use the `star`(1M) command to restore the damaged and nonexistent files identified in Step c and Step d.
7. Reimplement disaster recovery scripts, methods, and `cron`(1M) jobs using information from the backup copies.

Recreating a File System

Recreating a File System

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

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

This section describes this procedure.

How to Back Up and Re-create a File System

1. Back up all site-customized system files and configuration files.
   Depending on your software, these files might include `mcf`, `archiver.cmd`, `defaults.conf`, `samfs.cmd`, or `inquiry.conf`. Back up these files for all file systems in your Sun QFS environment. Also make sure that you have backup copies of files in the `/etc/opt/SUNWsamfs` directory, files in the `/var/opt/SUNWsamfs` directory, and shared hosts files.
2. Ensure that each file system to be modified is backed up.
   File systems should be backed up regularly according to your site's policies. If you are comfortable with the backup files that already exist for your file systems, there is no need to back them up again now. If, however, you need to back up your file systems to preserve information created since the last dump file was created, do so now. For information about how to create a dump file using `qfsdump`, see Backing Up SAM-QFS Data and Files.
3. Unmount the file system.
   For instructions, see Unmounting a File System.
4. If you want to rename the file system during this procedure, use the `samfsck`(1M) command with its `-R` and `-F` options.
   For more information, see the `samfsck`(1M) man page.
5. Edit the `/etc/opt/SUNWsamfs/mcf` file to add, change, or remove partitions.
   For more information, see Adding Disk Cache to a File System.
6. Type the `sam-fsd`(1M) command to check for errors in the `mcf` file:

```
# sam-fsd
```

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

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

```
# samd config
```

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

8. Issue the `sammkfs`(1M) command to re-create the file system.
   For example, the following command creates `samfs10`:

```
# sammkfs samfs10
```

9. Issue the `mount`(1M) command to mount the file system.
   For information about mounting a Sun QFS file system, see the `mount_samfs`(1M) man page.
10. Issue the `cd`(1) command to change to the mount point of the file system.
11. Use the `qfsrestore`(1M) command, or use SAM-QFS Manager, to restore each file.
   Restore from the dump file you had or from the dump file created in Step 1.
   For more information, see the `qfsdump`(1M) man page or the SAM-QFS Manager online help.

Restoring Files and Directories

Contents

Restoring Files and Directories

This section describes how to restore individual files and directories.

> Note -
> If the mv(1) command has been used to move an archived file to a different directory, the file is not rearchived. If you use the star(1M) command to recover a moved file, the star(1M) header on the archive media retains the original path name. When you use the star(1M) command to reload the file, the file is restored to its original location.
> You can see the path by issuing the star(1M) command with its tvbf arguments. Then, you can extract the file to its original location by issuing the star(1M) command again. Finally, issue the mv(1) command to move the file to its new directory.

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

You can use either of the following procedures to restore Sun StorageTek SAM or SAM-QFS files and directories that were archived to tape or magneto-optical cartridges. These procedures use the dump file created by samfsdump(1M).

Beginning with SAM-QFS Manager version 2.1, compressed metadata snapshots created by SAM-QFS Manager can be indexed without being uncompressed. In order to take advantage of this feature, you should select the gzip compression method for any scheduled metadata snapshots.

If you have existing compressed snapshots that are not in the gzip format, you can use the gznew command to convert them to gzip format.

In addition, indexing for metadata snapshots was also improved in version 2.1 of SAM-QFS Manager. Additional information was added to the index, including information about damaged or online files. To take advantage of this improvement, you should delete any existing indexes and recreate them.

How to Restore Files Using SAM-QFS Manager

1. From the Servers page, click the name of the server that contains the file system that you want.
   The File Systems Summary page is displayed.
2. Navigate to the File Browsing & Recovery node in the navigation tree.
   The File Browser page is displayed.

3. Use the file system mount point down menu to choose the file system you want to restore.
   A set of radio buttons displays on the top right corner of the table: Live Data and Recovery Point.

> Note -
> If you do not see these radio buttons, the selected recovery point has not been indexed. You must index the recovery point on the Recovery Points page before you can browse the recovery point file within the File Browser.

4. Select the Recovery Point radio button and choose a date of the recovery point you want to restore.
   The contents of the selected recovery point are displayed.
5. Locate the file you want to restore by browsing within the File Browser.
6. Select Restore from the Operations drop-down menu.
   The Restore window is displayed.
7. Specify the location to which you want to restore the file or directory.
   By default, the location is the path of the original file or directory, relative to the mount point of the file system. You can specify a different path relative to the mount point, or you can specify an absolute path on any archiving file system.
8. From the Online Status After Restoring drop-down menu, choose the actions that you want the file system to take after completing the restore process.
9. Click Submit.

How to Restore Files Using a **samfsdump**(1M) File

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

1. Use the `mkdir`(1) command to create the directory to which you want to restore the files within a SAM-QFS file system.
   # **`mkdir restore`**
2. Use the `archive`(1) command with the `-r` and `-n` options to prevent the archiver from archiving from this temporary directory location.

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

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

```
# cd restore
```

4. Use the `samfsrestore`(1M) command with the `-t` and `-f` options to list the contents of the dump file. After the `-f` option specify the dump file's path name, as shown in the following example.

```
# samfsrestore -t -f /dump_sam1/041126
samfsrestore -t -f /dump_sam1/041126
./lost+found
./neptune
./mary
./fileA
./fileB
./fileC
./fileD
./fileE
./mary/mary1
./mary/mary2
./neptune/vmcore.0
./neptune/unix.0
./neptune/bounds
```

5. Search the listing from the previous step to verify that the lost file is in the dump file.
   If you find the file you are looking for, copy down the exact path name shown in the output to use in the following step.
   In the previous screen example, the lost file, called `mary1`, is shown as residing in the `./mary` directory.
6. Use the `samfsrestore (1m)` command with the `-T` and `-f` options to restore the file's inode information to the current directory.

The file name you specify must match exactly the path name as it was listed in the previous output. The following example shows the use of `samfsrestore` to retrieve the file `./mary/mary1` from the dump file `/dump_sam1/041126`.

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

7. Use the `sls`(1) command with the `-D` option to list detailed information about the file, and verify that the inode information for the correct file has been retrieved.
   The following example shows inode information for file `./mary/mary1`.

```
# sls -D ./mary/mary1
mary/mary1:
  mode: -rw-rw----  links:   1  owner: mary     group: sam
  length:        53  inode:          43
  offline;  archdone;
  copy 1: ---- Nov 17 12:35          8ae.1 xt 000000
  copy 2: ---- Nov 17 15:51       cd3.7f57 xt 000000
  access:      Nov 17 12:33  modification: Nov 17 12:33
  changed:     Nov 17 12:33  attributes:   Nov 17 15:49
  creation:    Nov 17 12:33  residence:    Nov 17 15:52
```

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

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

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

The following table lists the procedures used to restore various types of files when no `samfsdump`(1M) output is available.

Restoring Files When No **sampfdump**(1M) Output Is Available

Type of File	Condition	Where Described
Regular file archived to removable media cartridges	An archiver log file exists with an entry for the file, or you have output from the `sls` command with the `-D` option that lists the file.	Restoring a Regular File Using Archiver Log or `sls` Information
Regular file archived to removable media cartridges	No archiver log file exists.	Restoring a Regular File Without Information From an Archiver Log
Regular file archived to disk	An archiver log file exists with an entry for the file, or you have output from the `sls` command with the `-D` option that lists the file.	Restoring Files Archived to Disk
Segmented file	An archiver log file exists with entries for the file.	Restoring a Segmented File Using Information From an Archiver Log
Volume overflow file	An archiver log file exists with entries for the file.	Restoring a Volume Overflow File Using Information From an Archiver Log

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

- Restoring a Regular File Using Archiver Log or `sls` Information
- Determining the File Type

Note -
If you restore any type of file (regular file, segmented file, and so on) without samfsdump(1M) output, you re-create the .inodes file, and you lose the content of the original .inodes file, as well as any file attributes modified with the chmod(1), chown(1), or other command. The files are restored with their default attributes.

Determining the File Type

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

Regular File

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

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

Segmented File

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

Each segmented file has multiple entries in an archiver log. Example 5-1 shows three entries for segmented file seg/aaa. Field 12 has a S indicating that the file type is file segment.

Example 5-1 Archiver Log Entry for a Segmented File

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

Volume Overflow File

A volume overflow file is a file that is written on multiple volumes. A volume overflow file has multiple entries in an archiver log, one for each section of the file. Example 5-2 shows two entries for the two sections of the regular file big2d. Field 5 shows that the file starts on VSN CFX600 and overflows to VSN CFX603, and field 13 shows the section numbers, 0 and 1.

Example 5-2 Archiver Log Entry for a Volume Overflow File

```
A 2001/10/31 09:47:29 lt _CFX600_ arset1.1 3668e.1 samfs9 71950.15
2011823616 testdir1/big2d f _0_ 43
A 2001/10/31 09:47:29 lt _CFX603_ arset1.1 3844a.0 samfs9 71950.15
1209402048 testdir1/big2d f _1_ 41
```

Restoring a Regular File Using Archiver Log or **sls** Information

The following table shows the information you need from the archiver log or sls -D command output in order to restore a regular file.

Information Needed for Restoring a Regular File

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

If you can obtain the needed information about a regular file either from its archiver log entry or from output from the `sls(1)` command with the `-D` option, you can restore the file with the `request(1M)` and `star(1M)` commands. As shown in the examples that follow, you use the `request` command first to create a file whose contents represent the contents of one or more pieces of removable media. This new file is sometimes referred to as a request file. You then use the `star` command to extract the file.

How to Restore a Regular File Using Information From an Archiver Log or **sls** Command Output

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

1. Log in as, or switch user to, `root`.
2. Find and record the media type, the file's position, and the VSN.
3. If you have an archiver log, use `cat(1M)` or another command to search the archiver log file for an entry for the missing file.
   The following example shows a sample entry for a file archived on a tape followed by a sample entry for a file archived on an optical disk.

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

   For definitions of the relevant fields in the archiver log file, see Information Needed for Restoring a Regular File.

4. If you have output from the `sls` command with the `-D` option about the missing file, search that output. The following example shows output from this command for file `tape_test/file4`.

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

5. Record the media type, the file's position, and the VSN to use as input to the `request(1M)` command in the next step.
6. Use the `request(1M)` command with the `-p` option, followed by the hexadecimal 0x and the position number from the archiver log to position to the beginning of the `tar(1)` header for the file.

   > Note -
   > VSNs specified with the `request(1M)` command must reside on a local automated library.

   The following example creates a request file with the contents of the archive containing the example file from step 2a that is on tape:

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

   The following example creates a request file with the contents of the example file from step 2a that is on optical disk:

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

7. Use the `star`(1M) command to extract the files.
   The `star`(1M) command restores all the files from the archive file that you are pointing to with the request file.
   If you labeled the tape with a block size other than the default (16 kilobytes), you would use the block size in bytes divided by 512 (in place of the value 32) for the `star` command's `-b` option. You can see the tape block size by mounting the tape and observing either the `samu`(1M) utility's `t` display, the `samu` utility's `v` display (press CTRL-i for detail lines), or the output of the `dump_cat`(1M) command.

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

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

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

> Note -
> You can ignore the directory checksum error.

8. Use the `sls`(1) command to verify that you have extracted the lost file.
   The following example shows the command output for the file on the optical disk.

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

Restoring a Regular File Without Information From an Archiver Log

If you do not have an archive log available with an entry for a regular file, you can still restore the file using either an automated library or a manually mounted, stand-alone drive, under the following conditions:

- If you are using an automated library, the automated library daemon is active on the system.
- If you are using a manually mounted, stand-alone drive, `/kernel/drv/st.conf` is correctly configured for the tape drive that you are using. For more information about performing this task, see how to add tape support to the `st.conf` file in the Sun StorageTek Storage Archive Manager Installation and Upgrade Guide.

How to Restore a Regular File Without Information From an Archiver Log

> Note -
> If the only resources available consist of a cartridge containing archive copies and a Solaris system without SAM-QFS software installed, start this procedure with Step 3

1. If you are using an automated library, prevent the SAM-QFS software from using the tape drive.

    > Note -
    > If you are using a manually mounted, stand-alone drive, skip this step.

    You can use the `samu`(1M) command with the :`unavail` eq option, the `samcmd`(1M) command with the `unavail` eq option, the `devicetool`(1M) commands, or the `libmgr`(1M) command. For the `samu` and `samcmd` commands, specify the Equipment Number of the drive as eq. The Equipment Number for each device is specified in the `mcf` file.
    The following example shows the use of the `samcmd` command.

    ```
    # samcmd unavail 51
    ```

2. If you are using an automated library, use the `samload`(1M) command to load the desired volume into the drive.

    > Note -
    > If you are using a manually mounted, stand-alone drive, skip this step.

    For the command-line options to use, see the `samload`(1) man page. The following example shows the use of the `samload` command to load the cartridge that is in slot 3 of library 50 into the drive with Equipment Number 51.

    ```
    # samload 50:03 51
    ```

3. Use the `mt`(1M) command to rewind the tape.
    The following example shows this command applied to tape drive `/dev/rmt/2`.

    ```
    # mt -f /dev/rmt/2cbn rewind
    ```

    > Note -
    > Because the device name used in these examples ends with the `n` (no rewind) option, each of the commands in the following steps examines the next file on the tape.

4. Use od(1M) or another command to examine the ANSI label on the cartridge, and find the line that starts with `0000240`.
    The first file on the cartridge is the ANSI label. In the following example, the information you are looking for appears on the line that starts with `0000240`.

```
# od -c /dev/rmt/2cbn
0000000   V   O   L   1   X   X   X
0000020                                   S   A   M   -   F   S       1
0000040   .   0
0000060
0000100                                                               4
0000120   H   D   R   1
0000140                                               0   0   0   1   0
0000160   0   0   1   0   0   0   1   0   0       2   4   9   0   9
0000200                                                   S   A   M   -
0000220   F   S       1   .   0
0000240   H   D   R   2       1   6   3   8   4               1
0000260                                                   2   0   g 031
0000300
*
0000360
```

5. Note the five characters that appear after `H D R 2` on the line that starts `0000240`.
   These five characters are the bottom five digits of the block size, in decimal. In the previous screen example, the characters are `1 6 3 8 4`.
6. Use the bottom five digits to determine the block size used on the media.
   The following table shows the block sizes corresponding to these digits for the `dd`(1M) and `tar`(1) commands.

Bottom Five Digits of Block Size	Block Size for **dd**(1)	512-byte Blocks for **tar**(1) and **star**(1M)
16384	16 kilobytes	32 blocks
32768	32 kilobytes	64 blocks
65536	64 kilobytes	128 blocks
31072	128 kilobytes	256 blocks
62144	256 kilobytes	512 blocks
24288	512 kilobytes	1024 blocks
48576	1024 kilobytes	2048 blocks
97152	2048 kilobytes	4096 blocks

7. Issue one of the following commands:
   - If the `star`(1M) command is available, use it to find the file in the archive.
     Issue it with the number of 512-byte blocks obtained in the previous two steps.
     You can download the `star` command from a Sun StorageTek SAM system onto any Solaris system.

     > Note -
     > `star` files have an extended maximum file size of 1 terabytes-1. `tar` and `star` files have compatible formats only at file sizes less than or equal to 8 gigabytes-1. At file sizes larger than 8 gigabytes, the formats of `star` and `tar` files are not compatible. Therefore, you must use the `star` command to read archives larger than 8 gigabytes-1.

     The following code example shows the `star` command being used to examine the first `tar` file. The block size for both the `star`(1M) and `tar`(1) commands is specified in units of 512-byte blocks. The number 32 used after `-b` in the example is the number of 512-byte blocks that corresponds to the number 16384 in the ANSI label in Step 4, from the table in Step 6.

```
# star -tv -b 32 -f /dev/rmt/2cbn
-rw-rw----  0/1   102564 Sep  6 13:02 1996 test
6+1 records in
11+1 records out
```

     The following code example shows the same command used to examine the next `tar`(1) file.

```
# star -tv -b 32 -f /dev/rmt/2cbn
-rw-rw----  0/1   102564 Sep  6 13:02 1996 test
6+1 records in
11+1 records out
```

The following code example shows two copies of another file being examined.

```
# star -tv -b 32 -f /dev/rmt/2cbn
-rw-rw----  0/1   102564 Sep  6 13:02 1996 test2
6+1 records in
11+1 records out
# star -tv -b 32 -f /dev/rmt/2cbn
-rw-rw----  0/1   102564 Sep  6 13:02 1996 test2
6+1 records in
11+1 records out
```

The following code example shows that the end of the tape has been reached.

```
# star -tv -b 32 -f /dev/rmt/2cbn
0+0 records in
0+0 records out
tar: blocksize = 0
# mt -f /dev/rmt/2cbn status
Other tape drive:
 sense key(0x13)= EOT   residual= 0   retries= 0
 file no= 5   block no= 0
```

- If the star(1M) command is not available, use the dd(1M) and tar(1) commands to examine the archives.
  The following code example shows the dd command being used to examine the first tar file. The value 16k used for the input block size (ibs=) is the number in Step 6 that corresponds to the number 16384 in the ANSI label.

```
# dd if=/dev/rmt/2cbn ibs=16k obs=10k conv=sync | tar tvf -
-rw-rw----  0/1   102564 Sep  6 13:02 1996 test
6+1 records in
11+1 records out
```

The following code example shows the same command examining the next tar(1) file.

```
# dd if=/dev/rmt/2cbn ibs=16k obs=10k conv=sync | tar tvf -
-rw-rw----  0/1   102564 Sep  6 13:02 1996 test
6+1 records in
11+1 records out
```

The following code example shows the examination of two copies of another file.

```
# dd if=/dev/rmt/2cbn ibs=16k obs=10k conv=sync | tar tvf -
-rw-rw----  0/1   102564 Sep  6 13:02 1996 test2
6+1 records in
11+1 records out
# dd if=/dev/rmt/2cbn ibs=16k obs=10k conv=sync | tar tvf -
-rw-rw----  0/1   102564 Sep  6 13:02 1996 test2
6+1 records in
11+1 records out
```

The following code example shows that the end of the tape has been reached.

```
# dd if=/dev/rmt/2cbn ibs=16k obs=10k conv=sync | tar tvf -
0+0 records in
0+0 records out
tar: blocksize = 0
# mt -f /dev/rmt/2cbn status
Other tape drive:
 sense key(0x13)= EOT   residual= 0   retries= 0
 file no= 5   block no= 0
```

Note -
You might receive errors during this process. The following error indicates that the block size you selected does not match that of the tape:{{read: not enough space}}If you receive this error, correct the block size and try again.

8. When you find the missing file in an archive, extract the file by using either the `-x` option with the `star` command or the dd command with the `tar` command.
   The following code example shows these commands.

Note -
You can ignore the `dd: read error` in the first line of dd command output.

```
# dd if=/dev/samst/c0t1u0 bs=1k iseek=3374 of=/tmp/junk count=10
dd: read error: I/O error
8+0 records in
8+0 records out
# tar xvf /tmp/junk
# star -xv -f /tmp/junk
tar: blocksize = 1
-rw-rw---- 0/1 2673 May 1 15:41 1996 dir3/dir2/file0
-rw-rw---- 0/1 946 May  1 15:41 1996 dir3/dir1/file1
-rw-rw---- 0/1 468 May  1 15:41 1996 dir1/dir3/file0
```

Restoring a Segmented File Using Information From an Archiver Log

When a segmented file is archived or staged, it is archived and staged in chunks. Each segmented file has multiple entries in an archiver log.

If you can find entries for a missing segmented file in an archiver log, you can use the file's position, segment size, VSN, and media type to restore the file with the `request`(1M) and `star`(1M) commands.

Note -
In the following procedures, all segments are in the same tape `tar`(1) file, and no segment is overflowed. If your file has segments on more than one `tar`(1) file, you must use a separate `request`(1M) command for each `tar`(1) file position. If any segments are volume overflow files, use the procedure described in Restoring a Volume Overflow File Using Information From an Archiver Log for those segments.

How to Restore a Segmented File Using Information From Archiver Log Entries

Note -
You must have free space in the file system equal to two times the size of the file to be recovered.

1. Find the archiver log entries for the segmented file by the file system name (from field 8) and file name (from field 11).
   The following code example shows entries for segmented file `file2` in the `archiver.log` file.

```
A 2002/11/19 14:01:47 ib E00000 all.1 1276a.1 _samfs4_ 14.5 10485760 _seg/aaa/1_ S 0 51
A 2002/11/19 14:04:11 ib E00000 all.1 1276a.5002 _samfs4_ 15.5 10485760 _seg/aaa/2_ S 0 51
A 2002/11/19 14:06:24 ib E00000 all.1 1933a.1 _samfs4_ 16.5 184 _seg/aaa/3_ S 0 51
```

The code example above shows the archiver log file for file segments in file system `samfs4`. Each segment has its own entry and file name: `seg/aaa/1`, `seg/aaa/2`, and `seg/aaa/3`.

2. Do the following for each segment or group of segments that is located at a unique position, even if they are on the same VSN.
   If there are segments on different VSNs, make sure that you specify the correct media type and VSN for each.
3. Note the contents of fields in the archiver log.
   You will use information from the archiver log as input to the `request`(1M) command in Step 3 and to the `segment`(1) command in Step 9. The information you need is contained in the following fields:
   - Field 4. Media type on which the file is stored. For the supported media types, see the `mcf`(4) man page.
   - Field 5. VSN.
   - Field 7. File position. From the position indicator portion to the left of the period (`.`) in the field.
   - Field 10. Segment size. This is the length field.
     The first line in the previous screen example provides the following information:
   - The media type is `ib`.
   - The VSN is `E00000`.
   - The file's position is `1276a`.
   - The segment size is 10485760.
4. Issue the `request` (1M) command to create a removable media file that points to the segments.
5. request -m media-type -p 0x position-number -v VSN filename

> Note -
> VSNs specified in the `request`(1M) command must reside on a local automated library.

For example, the following command uses the values from the example lines in Step 1:

```
# request -m ib -p 0x1276a -v E00000 /sam3/rmfile
```

The preceding command retrieves the first two segments.

6. Issue the `star`(1M) command.
   Use the name of the file created in the previous step to read the segments from tape onto the disk, as shown in the following example.

```
# star xvbf 512 /sam3/rmfile
seg/aaa/1
seg/aaa/2
```

7. `stet`(1) command to change into the directory where the segmented files reside.
   The following example shows segmented files 1, 2, and 3 in the `seg/aaa` directory.

```
# cd seg
# pwd
/sam3/seg
# ls&ndash; l
total 8
drwxrwx---   2 root     other       4096 Jun 15 17:10 aaa/
# ls -l aaa
total 40968
-rw-rw----   1 root     other   10485760 Jun 15 17:06 1
-rw-rw----   1 root     other   10485760 Jun 15 17:06 2
-rw-rw----   1 root     other        184 Jun 15 17:07 3
# pwd
/sam3/seg
# cd aaa
# pwd
/sam3/seg/aaa
```

8. Use the `ls`(1) and `sort`(1) commands to list and sort the numbered files in numerical order, and use the `cat`(1M) command to join the files.
   The temporary file created in this step is not segmented.

```
# ls | sort -n | xargs cat > ../bbb
```

9. Use the `cd`(1) command to change to the directory above where the numbered files reside, and then use the `rm`(1) command to remove the numbered files.

```
# cd ..
# pwd
/sam3/seg
# ls&ndash; l
total 41000
drwxrwx---   2 root     other       4096 Jun 15 17:10 aaa/
-rw-rw----   1 root     other    20971704 Jun 15 17:11 bbb
# ls -l aaa
total 40968
-rw-rw----   1 root     other    10485760 Jun 15 17:06 1
-rw-rw----   1 root     other    10485760 Jun 15 17:06 2
-rw-rw----   1 root     other        184 Jun 15 17:07 3
# rm -rf aaa
```

10. Use the `touch`(1M) command to create an empty file.

```
# touch aaa
```

11. Use the `segment`(1) command to set the segment attribute on the file that you just created.

```
# _segment -l_ _segment-length_ _m_ _filename_
```

    where:

    - segment-length is the length of the segment in megabytes.
    - filename is the name of the file that you just created.
      To obtain the segment length, take the segment size from field 10 of the archiver log file entry and divide it by 1048576. For example, the segment size in the archiver log entry example in Step a is 10485760. Dividing the segment size by 1048576 yields 10 megabytes, which is entered as `10m` in the following example.

      ```
      # segment -l 10m aaa
      ```

12. Copy the temporary file created in Step 4 into the empty file created in Step 9, and then remove the temporary file, as shown in the following example.

```
# cp bbb aaa
# rm bbb
```

13. Issue the `sls`(1) command with the `-2K` option to list the segments of the segmented file in two lines of output.

```
# sls -2K aaa
-rw-rw----   1 root     other     20971704 Jun 15 17:12 aaa
---------- ----- sI {3,0,0,0}
-rw-rw----   1 root     other     10485760 Jun 15 17:12 aaa/1
---------- ----- sS
-rw-rw----   1 root     other     10485760 Jun 15 17:12 aaa/2
---------- ----- sS
-rw-rw----   1 root     other          184 Jun 15 17:12 aaa/3
---------- ----- sS
```

Restoring a Volume Overflow File Using Information From an Archiver Log

A volume overflow file is a file that is written on multiple volumes. If you can find entries for a missing volume overflow file in an archiver log, you can use the file's position, segment size, VSN, and media type to restore and reassemble the file with the `request`(1M), `star`(1M), dd (1M), and `cat`(1) commands.

How to Restore a Volume Overflow File Using Information From an Archiver Log

> Note -
> Free space must be available in the file system equal to two times the size of the file to be recovered.

1. Use `vi`(1M) or another command to examine the archiver log file that contains an entry for the file you are trying to recover.
   The following code example shows the archiver log file for `file3`, a sample volume overflow file.

   ```
   A 2004/08/23 10:28:51 sg 700036 ReleasePercent.1 12d55.1 qfs2
   11731.1 89128448 ReleasePercent/huge2/dir24/file3 f 0 210
   A 2004/08/23 10:28:51 sg 700034 ReleasePercent.1 15f9e.0 qfs2
   11731.1 525271552 ReleasePercent/huge2/dir24/file3 f 1 220
   ```

   The file is identified as a volume overflow file with two sections because the `f` in the third-to-last field indicates that the entry is for a regular file, and the `0` and the `1` in the second-to-last fields are section numbers. The fifth field shows that the file starts on VSN `700036` and overflows to information about `700034`.

2. Use the `request`(1M) command to create a removable media file that points to each section of the volume overflow file, as shown in the following example.

   ```
   # request -p 0x12d55 -m sg -v 700036 /samfs1/tp1
   # request -p 0x15f9e -m sg -v 700032 /samfs1/tp2
   ```

3. Use the `cd`(1M) and `dd`(1M) commands to recover the sections.
   Repeat the `dd`(1M) command for each remaining section.
   In the following example, a block size of 256 kilobytes is assumed for both tapes.

   ```
   # cd /qfs2
   # dd if=/samfs1/tp1 of=file3.0 ibs=256k
   340+0 records in
   174080+0 records out
   # dd if=/samfs1/tp2 of=file3.1 ibs=256k
   2004+0 records in
   1026048+0 records out
   ```

4. Use the `ls`(1M) command to examine the output and ensure that all pieces of the file are on the disk.

```
# ls -l file3.*
-rw-r--r--   1 root     other   89128960  Aug 31 12:07 file3.0
-rw-r--r--   1 root     other   525336576 Aug 31 12:14 file3.1
```

5. Use the `cat`(1M) and `star`(1M) commands to reassemble the file.

```
# cat file3.0 file3.1 > file3.2
# ls -l file3.*
-rw-r--r--   1 root     other    89128960  Aug 31 12:07 file3.0
-rw-r--r--   1 root     other    525336576 Aug 31 12:14 file3.1
-rw-r--r--   1 root     other    614465536 Aug 31 12:21 file3.2
# _star xvbf 256 file3.2_
ReleasePercent/huge2/dir24/file3
# _sls -D ReleasePercent/huge2/dir24/file3_
ReleasePercent/huge2/dir24/file3:
  mode: -rw-r--r--  links:   1  owner: root      group: other
  length: 614400000  admin id:      0  inode:    12481.1
  access:      Aug 31 12:40  modification: Aug 20 14:28
  changed:     Aug 31 12:43  attributes:   Aug 31 12:40
  creation:    Aug 31 12:40  residence:    Aug 31 12:40
```

Restoring Files Archived to Disk

The following sections tell you how to gather necessary information and then restore files archived to disk.

How to Gather Information for a Disk Archive Restoration

Before You Begin

You must have the following information before you can restore any files that have been archived to disk:

- The disk volume name
- The path to the disk archive `tar`(1) file
- The path name defined for the disk volume name in the `diskvols.conf` file

You must have either the saved `sls`(1) output or the archiver log files that include the files you want to restore.

Steps

1. Find the disk volume name and the path to the disk archive `tar`(1) file that contains the archived file.
   You can use either the `sls`(1) command with the `-D` option or the archiver log file entry.
   - You can use the `sls` method if you have `sls`(1) output for the files you want to restore.
     Examine the lines that correspond to the disk archive copies, indicated by a media type of `dk` in the fifth field. The second-to-the-last field in these lines shows the disk volume name as defined in the `diskvols.conf` file. The last field shows the path to the disk archive `tar`(1) file.
     In the following example, both the commands you type in and the information you are looking for are in bold text.

```
# _sls -D filea fileb filec_
filea:
  mode: -rw-r--r--  links:   1  owner: root      group: other
  length:     65732  admin id:      0  inode:   120235.783
  archdone;
  copy 1: ---- Nov  3 14:46     81366.1    dk _DISK_01_ _d8/d19/f102_
  copy 2: ---- Nov  3 14:54     2ec7e.209  dk _DISK_02_ _d2/d236/f126_
  copy 3: ---- Nov  3 14:58        bf.209  dk _DISK_03_ _f191_
  copy 4: ---- Nov  3 15:05      ea7a.209  lt 000064
  access:      Nov  3 14:35  modification: Nov  3 14:35
  changed:     Nov  3 14:35  attributes:   Nov  3 14:35
  creation:    Nov  3 14:35  residence:    Nov  3 14:35
fileb:
  mode: -rw-r--r--  links:   1  owner: root      group: other
  length:     65732  admin id:      0  inode:   120300.783
  archdone;
  copy 1: ---- Nov  3 14:46     81366.105  dk _DISK_01_ _d8/d19/f102_
  copy 2: ---- Nov  3 14:54     2ec7e.411  dk _DISK_02_ _d2/d236/f126_
  copy 3: ---- Nov  3 14:58        bf.411  dk _DISK_03_ _f191_
  copy 4: ---- Nov  3 15:05      ea7a.411  lt 000064
  access:      Nov  3 14:35  modification: Nov  3 14:35
  changed:     Nov  3 14:35  attributes:   Nov  3 14:35
  creation:    Nov  3 14:35  residence:    Nov  3 14:35
.
.
.
```

- You can use the archiver log method if you have an archiver log file that includes the files you want to restore.
  Examine the lines corresponding to the disk archive copies, indicated by a media type of dk in the fourth field. The fifth field in these lines shows the disk volume name as defined in `diskvols.conf`, and a slash (/) character, and then the path to the disk archive `tar`(1) file.
  In the following example, the information you are looking for is in bold text.

```
A 2003/11/03 14:46:35 dk _DISK_01/d8/d19/f102_ arset4.1 81366.1 shareqfs2 120235.783
65732 testdir4/filea f 0 0
A 2003/11/03 14:46:35 dk _DISK_01/d8/d19/f102_ arset4.1 81366.83 shareqfs2 120243.783
65732 testdir4/filec f 0 0A 2003/11/03 14:46:35 dk _DISK_01/d8/d19/f102_ arset4.1
81366.105 shareqfs2 120300.783 65732 testdir4/fileb f 0 0
A 2003/11/03 14:50:35 dk _DISK_01/d8/d19/f103_ arset4.1 81367.3 shareqfs2 120228.783
131420
A 2003/11/03 14:54:35 dk _DISK_02/d2/d236/f126_ arset4.2 2ec7e.38f shareqfs2
120243.783 65732 testdir4/filec f 0 0
A 2003/11/03 14:54:35 dk _DISK_02/d2/d236/f126_ arset4.2 2ec7e.411 shareqfs2
120300.783 65732 testdir4/fileb f 0 0
A 2003/11/03 14:58:35 dk _DISK_03/f191_ arset4.3 bf.3 shareqfs2 120228.783 131420
.
.
.
```

2. Use `cat`(1) or another command to examine the `diskvols.conf` file and to find the path name defined for the disk volume name in `diskvols.conf`.
   In the following example, two of the three disk volumes defined for receiving disk archive copies are mounted locally, and one is mounted remotely, on server `mars`.

```
# cat /etc/opt/SUNWsamfs/diskvols.conf
DISK_01 /ufs2/disk_archive/01
DISK_02 /ufs2/disk_archive/02
DISK_03 mars:/qfs1/disk_archive/03
```

How to Restore Files From a Disk Archive **`tar`**(1) File

Before you start this procedure, collect the information described in Restoring Files Archived to Disk.

1. Use the mkdir(1) command to create the directory in which you will restore the files.
2. Use the cd(1) command to change to the restoration directory.
3. Use the star(1M) command with its -tv option to list the content of the disk archive tar(1) file.
   The following example shows the content of the disk archive tar(1) file associated with archive copy 1.

```
# star -tv -f /ufs2/disk_archive/01/d8/d19/f102
-rw-r--r-- root/other    65732 2003-11-03 14:35 testdir4/filea
-rw-r--r-- root/other    65732 2003-11-03 14:35 testdir4/filec
-rw-r--r-- root/other    65732 2003-11-03 14:35 testdir4/fileb
```

   Note -
   If the tar file is on a remote server, accessing it requires proper configuration of the remote authentication database. For information on configuring the /.rhosts file, see the hosts.equiv(4) man page.

4. Verify that the files that you want to restore are listed in the output from Step 3.
   If you are restoring a single file but do not know its exact path name, use this information in the next step.
5. Use the star(1M) command with its -xv option to restore the files.
   The following example shows how the star(1M) command is used to retrieve the single file testdir4/fileb from disk archive tar(1) file /ufs2/disk_archive/01/d8/d19/f102.

```
# star -xv -f /ufs2/disk_archive/01/d8/d19/f102 testdir4/fileb
testdir4/fileb
```

   In the following example, the star(1M) command is used to retrieve all files from disk archive tar(1) file /ufs2/disk_archive/01/d8/d19/f102.

```
# star -xv -f /ufs2/disk_archive/01/d8/d19/f102
testdir4/filea
testdir4/filec
testdir4/fileb
```

6. Use the sls(1) command with its -DR option to verify that you have extracted the proper files.
   The following example shows multiple retrieved files.

```
# sls -DR
testdir4:
  mode: drwxr-xr-x  links:   2  owner: root      group: other
  length:      4096  admin id:      0  inode:   120274.789
  access:      Nov  4 14:11  modification: Nov  4 14:11
  changed:     Nov  4 14:11  attributes:   Nov  4 14:11
  creation:    Nov  4 14:11  residence:    Nov  4 14:11
testdir4:
testdir4/filea:
  mode: -rw-r--r--  links:   1  owner: root      group: other
  length:     65732  admin id:      0  inode:   120293.787
  access:      Nov  4 14:11  modification: Nov  3 14:35
  changed:     Nov  4 14:11  attributes:   Nov  4 14:11
  creation:    Nov  4 14:11  residence:    Nov  4 14:11

testdir4/fileb:
  mode: -rw-r--r--  links:   1  owner: root      group: other
  length:     65732  admin id:      0  inode:   120281.783
  access:      Nov  4 14:11  modification: Nov  3 14:35
  changed:     Nov  4 14:11  attributes:   Nov  4 14:11
  creation:    Nov  4 14:11  residence:    Nov  4 14:11

testdir4/filec:
  mode: -rw-r--r--  links:   1  owner: root      group: other
  length:     65732  admin id:      0  inode:   120280.783
  access:      Nov  4 14:11  modification: Nov  3 14:35
  changed:     Nov  4 14:11  attributes:   Nov  4 14:11
  creation:    Nov  4 14:11  residence:    Nov  4 14:11
```

Retrieving Unarchived Files From File Systems

Unarchived files that resided within a SAM-QFS file system might not be recoverable after a system outage. The following list contains information that might help you to retrieve unarchived files:

- You can use the `sfind`(1M) command line to identify all files in a file system that are not archived. The following command finds all unarchived files associated with the `/sam1` mount point:

```
# sfind /sam1 ! -archived
```

- If the `samfsdump`(1M) method was used to dump and back up metadata, the `samfsrestore`(1M) command identifies files without archive copies and flags them as damaged.
- Sun StorageTek SAM log files cannot help you determine which files were not archived and were therefore lost between the last archiver run and the system outage. However, you can determine the files that might not have been archived by analyzing the `archiver.cmd` file for archiving directives and intervals. If all files are eligible for archiving, you can find the age of the oldest unarchived (lost) files in the `archiver.cmd` file's contents.
- You can use the `-l` and `-v` options with the `archiver`(1M) command to determine whether volumes were available to archive each archive set's data before the outage. Lack of sufficient volumes can prevent archiving of data in one or more archive sets. For information about the `archiver`(1M) command, see the `sam-archiverd`(1M) man page.
- If you are recovering files directly from a backup tape in `tar`(1) format, the files are restored to their locations according to the information on the tape. The path name is relative to the mount point of the file system. If any files have been moved within the system since the archive copies were created, they are restored to their original locations, not to their new locations.

Related Topics

- Salvaging Damaged Volumes
- Recovering File Systems
- Recovering From Catastrophic Failure

Salvaging Damaged Volumes

Contents

Salvaging Damaged Volumes

This chapter describes how to restore data from tapes or magneto-optical disks that are not usable in a SAM-QFS environment. This procedures in this chapter describe what to do when a volume is partially corrupted, was accidentally relabeled, has a destroyed label, or is entirely destroyed. The procedures in this chapter describe how to recover data both when archive copies are available and when there are no other copies available.

> Note -
> Before attempting the procedures in this chapter, use software other than Sun StorageTek SAM tools to determine whether the volume can be read. Try reading the volume in multiple drives, or try using the `}} {{ tar` (1) command.

Recovering Data From a Tape Volume

The procedures for recovering data from a tape volume vary, depending on the nature of the damage and whether additional archive copies of the volume's files are present on another tape. This section describes how to recover data in the following circumstances:

- The tape volume is damaged, and alternative archive copies are available.
- The tape volume is partially corrupt, and no alternative archive copies are available.
- The tape volume was accidentally relabeled, and no alternative archive copies are available.
- The Sun StorageTek SAM software cannot read the tape volume label, and no alternative archive copies are available.

Damaged Tape Volume With Other Copies Available

The Sun StorageTek SAM software allows you to create up to four archive copies of each online file. By default, only one copy is made, but you should make at least two copies to physically different archive media.

When an alternative archive copy is available, the recovery procedure includes a step for rearchiving all archive copies currently stored on the damaged volume before dispensing with the damaged volume. The new archive copies are made from the alternative archive copy.

How to Recycle a Damaged Tape With Other Copies Available

Use this procedure if alternative archive copies exist on volumes that are stored on site and are available for staging.

1. Export the damaged volume from the tape library, and flag it as unavailable in the historian catalog.
   Issue the `export`(1M) and `chmed`(1M) commands as shown in the following example, specifying the media type (mt) and volume serial number (vsn) of the damaged volume.

```
# export _mt_._vsn_
# chmed +U _mt_._vsn_
```

2. Flag the unavailable volume for recycling.
   Use the chmed(1M) command, and specify the media type (mt) and the VSN (vsn) of the damaged volume.

   ```
   # chmed +c _mt_._vsn_
   ```

3. Set the -ignore option for the library in the recycler.cmd file.
   The following example shows the -ignore option set on the lt20 library:

   ```
   # vi /etc/opt/SUNWsamfs/recycler.cmd
   logfile = /var/adm/recycler.log
   lt20 -hwm 75 -mingain 60 -ignore
   :wq
   ```

   For more information about the ignore option, see the recycler-cmd(4) man page.

4. Run the sam-recycler(1M) }} command with the {{-x option from the command line.

   ```
   # sam-recycler -x
   ```

   When the recycler runs, it does not select any volumes for recycling other than the volume that you have marked as unavailable. The recycler identifies all active archive copies on this volume and flags those archive copies for rearchiving. The next time the archiver runs, the archive copies marked for rearchiving are written to new volumes.
   After the archive copies have been written to new volumes, the damaged volume that you are recycling is considered to be drained of active archive copies.

5. Dispense with the volume.
   How you dispense with the volume depends on the nature of the damage. Use the following guidelines:
   - If the tape was accidentally relabeled, or if the tape label is unreadable, use the tplabel(1M) command to relabel the volume.
   - If relabeling the volume fails, export the volume from the historian and dispose of the tape.

   > Note -
   > If the tape is either partially corrupt or completely destroyed, do not reuse the tape VSN after the volume has been exported from the historian catalog.

Damaged Tape Volume With No Other Copies Available

If a tape volume is partially corrupt, you may be able to recover data from the parts of the tape volume that are not corrupt. This process is not an exact science, and it requires some trial and error to recover as much data as possible.

Errors logged in the device log can help you determine the area of a tape that is damaged. The archive_audit(1M) command can be used to generate the position and offset information for all archived files for a specific file system. You can use this position and offset information to help determine which archive copies are written to an area of a tape that is damaged.

How to Recover Files From a Damaged Tape With No Other Copies Available

1. Use the archive_audit(1M) command to generate a list of all files with archive copies on the partially corrupt tape volume.
   Issue the command as shown in the following example, specifying the file system's mount point, the VSN (vsn) of the volume, and an output file name.

   ```
   # archive_audit /_mount-point_ | grep _vsn_ > _filename_
   ```

2. Edit the output file from the archive_audit(1M) command to remove the lines for the files in the damaged area. Save the list of deleted files for inspection in Step 3.
3. Use the list of files with archive copies that cannot be accessed (the ones that are written in the area of the tape determined to be damaged) to determine whether any of the files are still on the disk.

Files that are not on disk cannot be recovered. You can remove these unrecoverable files from the file system.

4. Edit and run the `stageback.sh` script on the `archive_audit` output file you edited in Step 2.
   The `stageback.sh` script can stage each file from `archive_audit` output, set it to `no-release`, and mark the file for rearchiving{{.}}
   For information about the `stageback.sh` script, see Backup and Recovery Commands and Tools.
5. Open the `/opt/SUNWsamfs/examples/stageback.sh` file for editing.

```
# _cd /opt/SUNWsamfs/examples_
# _vi stageback.sh_
```

6. In the section that begins with `# echo rearch $file`, replace the word `media` with the media type (mt) and the word VSN with the VSN of the damaged volume, which are the same as the VSNs in Step 1.

```
# echo rearch $file
#
# Edit the following line for the correct media type and VSN
#
# eval /opt/SUNWsamfs/bin/rearch -m media -v VSN $file
```

7. Remove the pound sign from the beginning of the lines in the section shown in Step b.
   The file should now look like the following example.

```
echo rearch $file
# Edit the following line for the correct media type and VSN
eval /opt/SUNWsamfs/bin/rearch -m _media_ -v _VSN_ $file
```

8. Save and quit the file.
9. Run the `stageback.sh` script.

Relabeled Tape Volume With No Other Copies Available

The Sun StorageTek SAM software cannot read beyond the end of data (EOD). If a tape is accidentally relabeled, the only possibility for recovering data is to determine whether the tape manufacturer offers a method for reading beyond EOD.

If the tape manufacturer can provide a mechanism for reading beyond EOD, you can recover the data by combining that process with the procedure for recovering files from a tape volume with a label not readable by the Sun StorageTek SAM software. This procedure is described under Unreadable Tape Label With No Other Copies Available.

Unreadable Tape Label With No Other Copies Available

Whenever the Sun StorageTek SAM software receives a request to mount a tape volume in a drive, one of the first actions it takes is to verify the tape label. If the tape label cannot be read, the Sun StorageTek SAM software cannot use the tape for staging or archiving activities.

You can use the `tarback.sh` script to recover data from a tape with a label that cannot be read. The shell script automates the process of recovering data written to a tape by using the `star`(1M) command to read each archive file written on a tape volume. The file data is read back onto disk (into a Sun StorageTek SAM or UNIX file system) as data. File data recovered in this manner can then be moved to the appropriate location in the Sun StorageTek SAM file system and archived as new data.

How to Recover Files From a Tape Whose Label Is Unreadable

1. If you are using this process to recover file data from several tapes, disable any currently occurring recycling.
   When recycling is in process, data on the tape volumes may be inaccessible.
2. Use the `cp`(1M) command to copy the `tarback.sh` file to a working location, as shown in the following example.

```
# cp /opt/SUNWsamfs/examples/tarback.sh /var/tarback.sh
```

3. Issue the `samcmd`(1M) command with the `unavail` option to prevent the tape drive from being used for staging and archiving activities.

Type the Equipment Number value of the drive, as specified in the `mcf` file. For eq,

```
# samcmd unavail _eq_
```

4. Edit the working copy of the `tarback.sh` script to specify the variables shown in the following table.

Variable	Definition
EQ="eq"	The Equipment Number value of the tape drive as defined in the `mcf` file.
TAPEDRIVE=" path"	The raw path to the device that is described by EQ=.
BLOCKSIZE=" size"	The block size, in 512-byte units. Specify 256 for a block size of 128 kilobytes.
MEDIATYPE=" mt"	The two-character media type for this tape as defined in the `mcf`(4) man page.
VSN_LIST=" vsn1 vsn2 ..."	The list of VSNs to be read. There is no limit on the number of VSNs that can be specified. Use a space character to separate the VSNs.You can continue this list on another line by using a backslash (}}) `character. For example:{{VSN_LIST="vsn1 vsn2 }}{{vsn3"`

5. Execute the `tarback.sh` script.

Recovering Data From a Magneto-optical Volume

The procedures for recovering data from a magneto-optical volume vary, depending on the nature of the damage and whether additional archive copies of the volume's files are present on another tape. This section describes how to recover data in the following circumstances:

- The magneto-optical volume is damaged, and alternative archive copies are available.
  See Damaged Magneto-optical Volume With Copies Available.
- The magneto-optical volume is damaged, and no alternative archive copies are available.
  See Damaged Magneto-optical Volume With No Other Copies Available.
- The magneto-optical volume was accidentally relabeled, and no alternative archive copies are available.
  See Relabeled Magneto-optical Volume With No Other Copies Available.
- The Sun StorageTek SAM software cannot read the magneto-optical volume label, and no alternative archive copies are available.
  See Unreadable Label With No Other Copies Available.

Damaged Magneto-optical Volume With Copies Available

Regardless of the nature of the damage to the magneto-optical volume, if an alternative archive copy is available, you should use the good magneto-optical volume as your primary set of archive copies.

The recovery procedure includes a step for rearchiving all archive copies currently stored on the damaged volume before dispensing with the damaged volume. The new archive copies are made from the available alternative archive copy.

How to Rearchive Files and Recycle a Damaged Magneto-optical Volume With Copies Available

Use this procedure if readable alternative archive copies exist on volumes that are available on-site for staging.

1. Issue the `samexport`(1M) command to export the damaged volume from the magneto-optical library.
   Use the syntax shown in the following example, specifying the media type (mt) and VSN (vsn) of the damaged volume.

```
# samexport _mt_._vsn_
```

2. Issue the `chmed`(1M) command with the `-U` option to flag the damaged volume as unavailable in the historian catalog.
   Specify the media type (mt) and VSN (vsn) of the damaged volume.

```
# chmed +U _mt_._vsn_
```

3. Issue unavailable volume for recycling.
   Specify the media type (mt) and the VSN (vsn) of the damaged volume.

```
# chmed +c _mt_._vsn_
```

4. Edit the `recycler.cmd` file to set the `-ignore` option for the library.
   The following example shows the `-ignore` option set on the `lt20` library.

```
# vi /etc/opt/SUNWsamfs/recycler.cmd
logfile = /var/adm/recycler.log
lt20 -hwm 75 -mingain 60 -ignore
:wq
```

5. Enter the `sam-recycler`(1M) command with the `-x` option.

```
# sam-recycler -x
```

   When the recycler runs, it does not select any volumes for recycling other than the volume that you have marked as unavailable. The recycler identifies all active archive copies on this volume and flags those archive copies for rearchiving. The next time the archiver runs, the archive copies marked for rearchiving are written to new volumes.
   After the archive copies have been written to new volumes, the damaged volume that you are recycling is considered to be drained of active archive copies.

6. Dispense with the volume.
   How you dispense with the volume depends on the nature of the damage. Use the following guidelines:

- If the magneto-optical volume was accidentally relabeled, use the `odlabel`(1M) command to relabel the volume.
- If the magneto-optical label is unreadable, or if the magneto-optical volume is partially corrupt or completely destroyed, export the volume from the historian and dispose of it.

> Note -
> If the magneto-optical volume is either partially corrupt or completely destroyed, do not reuse the magneto-optical label after the volume has been exported from the historian catalog.

If the magneto-optical volume is completely destroyed and no alternative archive copies exist, there is no chance for recovering any data from this magneto-optical platter.

Damaged Magneto-optical Volume With No Other Copies Available

If a magneto-optical volume is only partially corrupt, it is possible to recover data written to the parts of the magneto-optical volume that are not damaged. This process requires some trial and error to recover as much data as possible.

It is possible to determine the area of a magnetic optical volume that is damaged from errors logged in the device logs. By using file names for files that cannot be retrieved, you can determine the location of the damage using the position and offset data.

The `archive_audit`(1M) command audits all archive copies for a specific file system. The output of the `archive_audit` command includes the position and offset information for each archive copy. You can use this position and offset information to help determine which archive copies are written to an area of a damaged magneto-optical disk.

How to Recover From a Damaged Magneto-optical Volume With No Other Copies Available

Copies of files that were archived outside the damaged area on a magneto-optical volume may be accessible. You can use the following procedure to recover files in accessible areas of a partially corrupted magneto-optical volume.

1. Issue the `archive_audit`(1M) command to generate a list of all files with archive copies on the partially corrupt tape volume.
   Use the syntax shown in the following example, specifying the file system's mount point, the VSN of the damaged volume, and an output file name.

```
# archive_audit /_mount-point_ | grep _vsn_ > _filename_
```

2. Edit the `archive_audit` output file and create three separate files with the following contents:
   - Files that appear before the damaged area on the magneto-optical disk
   - Files that appear within the damaged area
   - Files that appear after the damaged area
3. Look for the files with archive copies within the damaged area of the magneto-optical disk to determine whether any of the files are still in disk cache.
   Files that are not in disk cache cannot be recovered.
4. Remove the files identified as unrecoverable in Step 2 from the file system.
5. Edit and run the `stageback.sh` script using the files created in Step 2 that list files outside the damaged area.
   The `stageback.sh` script stages each file from `archive_audit` output, sets it to `no-release`, and marks the file for rearchiving.
   For information about the `stageback.sh` script, see Broken Link (Target ID: CHAP01.SGM).
6. Open the `/opt/SUNWsamfs/examples/stageback.sh` file for editing.

```
# _cd /opt/SUNWsamfs/examples_
# _vi stageback.sh_
```

7. In the section that begins with `# echo rearch $file` replace the word `media` with the media type and the word VSN with the same VSN specified in Step 1.

```
# echo rearch $file
#
# Edit the following line for the correct media type and VSN
#
# eval /opt/SUNWsamfs/bin/rearch -m media -v VSN $file
```

8. Remove the pound sign from the beginning of the lines in the section shown in Step .

```
echo rearch $file
# Edit the following line for the correct media type and VSN
eval /opt/SUNWsamfs/bin/rearch -m _media_ -v _VSN_ $file
```

9. Save and quit the file.
10. Run the `stageback.sh` script.

Relabeled Magneto-optical Volume With No Other Copies Available

Unlike tape media, magneto-optical media do not have an EOD marker. When a magneto-optical volume is accidentally relabeled, the Sun StorageTek SAM software cannot access data written previously. If the label date on the magneto-optical volume is newer than the archive copy date of files, that data is no longer accessible.

Contact Sun Microsystems customer support if a magneto-optical volume is accidentally relabeled. It is sometimes possible to recover some of this data with a special (but unsupported) `samst` driver that ignores the magneto-optical label date. This driver is not a standard part of the Sun StorageTek SAM product, and it is not released as part of the product. It can only be made available by Sun customer support.

Unreadable Label With No Other Copies Available

For magneto-optical media, there is no standard Solaris approach for locating and skipping to the various `tar`(1M) files. Contact Sun Microsystems customer support if you need to access files on a magneto-optical volume with an unreadable label.

Related Topics

- Recovering File Systems
- Recovering From Catastrophic Failure

Tools for SAM Troubleshooting

Contents

Tools for SAM Troubleshooting

The following sections provide an overview of some of the tools you might use when troubleshooting issues in the Sun Storage Archive Manager (SAM) environment

SAM Daemons

This section describes the daemons that can be present in a SAM environment and show how to verify the functionality of these daemons.

The process spawner, `init`(1M), starts the `sam-fsd`(1M) daemon based on information defined in `inittab`(4). The `sam-fsd`(1M) daemon provides overall control of the initialization of the SAM environment. As part of this process, it starts a number of child daemons. These child daemons are as follows:

- `sam-archiverd`(1M) - Controls the file archiving process in a Sun StorageTek SAM environment. The `sam-archiverd`(1M) daemon starts one `sam-arfind`(1M) process per mounted file system. In addition, the `sam-archiverd`(1M) daemon starts a variable number of `sam-arcopy`(1M) processes, depending upon the level of archiving activity and number of archive requests that are generated by the `sam-arfind`(1M) processes.
- `sam-stagerd`(1M) - Controls the file staging process. This daemon starts the `sam-stagerd_copy`(1M) processes, which copy archived files from archive media to the online disk cache.
- `sam-stagealld`(1M) - Controls the associative staging of files.
- `sam-ftpd`(1M) - Transfers data between local and remote Sun StorageTek SAM systems when Sun SAM-Remote is configured.
- `sam-amld`(1M) - Initializes several parts of the system and starts the following daemons as necessary:
  - `sam-scannerd`(1M) - Monitors all manually mounted removable-media devices. The scanner periodically checks each device for inserted archive media cartridges.
  - `sam-catserverd`(1M) - Builds and maintains library catalog files for automated libraries.
  - `sam-robotsd`(1M) - Starts and monitors the robot control daemons for automated libraries and media changers. The `sam-robotsd`(1M) daemon, in turn, starts various daemons, depending on the types of robots attached and whether they are direct attached or network attached.

The following table shows the daemons specific to various automated libraries.

Table - Automated Library Daemons

Daemon	Description
`sam-robotsd`	Monitors the execution of robot control daemons. The `sam-robotsd` daemon is started automatically by the `sam-amld` daemon.
`sam-genericd`	Controls direct attached libraries and media changers. Also controls ADIC libraries through the DAS interface.
`sam-stkd`	Controls the StorageTek media changers through the ACSAPI interface.
`sam-ibm3494d`	Controls the IBM 3494 tape libraries through the `lmcpd` interface.

sam-sonyd	Controls the Sony network attached automated libraries through the DZC-8000S interface.

Verifying SAM Daemons

It is possible to determine which daemons and processes should be running for a given configuration based on a knowledge of the Sun StorageTek SAM daemons and processes and the circumstances under which they are started. You can check that the expected daemons or processes are running by using the ps(1) and ptree(1) commands.

The following example assumes that the ps(1) command is issued in a Sun StorageTek SAM environment that includes a StorageTek L700 library connected by Automatic Cartridge System Library Software (ACSLS) to a Sun StorageTek SAM system with two mounted file systems, samfs1 and samfs2. In this example, the sam-stkd(1M) daemon is running. This controls the network attached StorageTek media changers through the ACSAPI interface implemented by the ACSLS software. If such equipment were present, similar daemons would be started for network attached IBM (sam-ibm3494d(1M)) and Sony (sam-sonyd(1M)) automated libraries, and for standard direct attached automated libraries that conform to the SCSI-II standard for media changers (sam-genericd(1M)).

```
skeeball # ps -ef | grep sam-fsd | grep -v grep
            root 656 1 0 10:42:26 ? 0:00 /usr/lib/fs/samfs/sam-fsd
skeeball # ptree 656
656 /usr/lib/fs/samfs/sam-fsd
        681 sam-archiverd
            931 sam-arfind samfs2
            952 sam-arfind samfs1
        683 sam-stagealld
        682 sam-ftpd
        684 sam-stagerd
        685 sam-amld
            687 sam-catserverd 1 2
            689 sam-scannerd 1 2
            690 sam-robotsd 1 2
                691 sam-stkd 1 2 30
                    692 /opt/SUNWsamfs/sbin/ssi_so 692 50014 23
                    694 sam-stk_helper 1 30
skeeball #
```

Checking **ps**(1) Output and Related Factors

Check the ps(1) command's output for missing or duplicate daemon processes and defunct processes. There should be only one of each of these processes, with few exceptions, as follows:

- One sam-arfind(1M) process per mounted file system.
- One sam-stkd, sam-ibm3494d, sam-sonyd, or sam-genericd process per automated library defined in the master configuration file (mcf). For more information, see the sam-robotsd(1M) man page.
- Zero or more sam-arcopy(1M) processes, depending on configuration and archiving load.
- Zero or more sam-stagerd_copy(1M) processes, depending on configuration and staging load.

The sam-fsd(1M) daemon reads the following configuration files: mcf, defaults.conf, diskvols.conf, and samfs.cmd. Verify that these configuration files are error free by issuing the sam-fsd(1M) command manually and watching for error messages.

Example - **sam-fsd**(1M) Output

As the following example shows, if sam-fsd(1M) encounters errors when processing these files, it exits without starting up the Sun StorageTek SAM environment.

```
skeeball # sam-fsd
6: /dev/dsk/c1t2d0s0    10   md    samfs1     on     /dev/rdsk/c1t2d0s0
 *** Error in line 6: Equipment Number 10 already in use
1 error in &rdquo;/etc/opt/SUNWsamfs/mcf&rsquo;
sam-fsd: Read mcf /etc/opt/SUNWsamfs/mcf failed.
skeeball #
```

Many of these files are described in the following sections:

- SAMQFS:The /etc/opt/SUNWsamfs/mcf File
- SAMQFS:The /kernel/drv/st.conf File

* SAMQFS:The /kernel/drv/samst.conf File
* SAMQFS:The /etc/opt/SUNWsamfs/inquiry.conf File
* SAMQFS:The /etc/opt/SUNWsamfs/defaults.conf File

Log and Trace Files

Using the appropriate log and trace files can greatly facilitate the diagnosis of Sun StorageTek SAM problems. Table - Log and Trace File Summary shows the relevant files.

Table - Log and Trace File Summary

File	Default Location
Sun StorageTek SAM log file	Configurable. Defined in /etc/syslog.conf.
System messages file	/var/adm/messages
Device logs	/var/opt/SUNWsamfs/devlog/eq
Daemon trace files	Configurable. Defined in /var/opt/SUNWsamfs/trace.
Archiver log file	Configurable. Defined in archiver.cmd.
Releaser log file	Configurable. Defined in releaser.cmd.
Stager log file	Configurable. Defined in stager.cmd.
Recycler log file	Configurable. Defined in recycler.cmd.

The following sections describe how to use the log and trace files when troubleshooting:

* SAMQFS:Enabling System Logging
* SAMQFS:Enabling Device Down Notification
* SAMQFS:Enabling Daemon Tracing
* SAMQFS:Enabling Device Logging

Enabling System Logging

The Sun StorageTek SAM software makes log entries using the standard Sun StorageTek SAM log file interface (see syslogd(1M), syslog.conf(4), syslog(3C)). All logging is done based on a level and a facility. The level describes the severity of the reported condition. The facility describes the component of the system sharing information with the syslogd(1M) daemon. The Sun StorageTek SAM software uses facility local7 by default.

How to Enable System Logging

To enable the syslogd(1M) daemon to receive information from the Sun StorageTek SAM software for system logging, perform the following steps:

1. Add a line to the /etc/syslog.conf file to enable logging.

   For example, add a line similar to the following:

   ```
   local7.debug /var/adm/sam-log
   ```

   You can copy this line from /opt/SUNWsamfs/examples/syslog.conf_changes. This entry is all one line, and it has a TAB character (not a space) between the fields.

2. Use touch(1) to create an empty /var/adm/sam-log file.
   For example:

   ```
   skeeball # touch /var/adm/sam-log
   ```

3. Send the syslogd(1M) process a SIGHUP signal.
   For example:

```
skeeball # ps -ef | grep syslogd | grep -v grep
    root   216     1  0   Jun 20 ?        0:00 /usr/sbin/syslogd
skeeball # kill -HUP 216
```

4. (Optional) Use `vi`(1) or another editor to open the `defaults.conf` file and add the debugging level.
   Perform this step only if you want to increase the logging level.
   You can use the `debug` keyword in the `defaults.conf` file to set the default level for the debug flags. These flags are used by the Sun StorageTek SAM daemons for logging system messages. The syntax for this line is as follows:

   ```
   debug = <option-list>
   ```

   The default debug level is `logging`, so `debug=logging` is the default specification. For option-list, specify a space-separated list of debug options. For more information on the options available, see the `samset`(1M) and `defaults.conf`(4) man pages.

Enabling Device Down Notification

The robot daemon, `sam-robotsd`(1M), starts and monitors the execution of the media changer control daemons in Sun StorageTek SAM systems. The `sam-amld`(1M) daemon automatically starts the `sam-robotsd`(1M) daemon if there are any media changers defined in the `mcf` file. For more information, see the `sam-robotsd`(1M) man page.
The `sam-robotsd`(1M) daemon executes the `/opt/SUNWsamfs/sbin/dev_down.sh` notification script when any removable media device is marked down or `off`. By default, it sends email to `root` with the relevant information. It can be tailored to use `syslogd`(1M) or to interface with the system management software in use at a site. For more information, see the `dev_down.sh`(4) man page.

Enabling Daemon Tracing

You can enable daemon tracing by configuring settings in the `defaults.conf` file. Example: Syntax to Enable Daemon Tracing for all Daemons shows the syntax to use in the `defaults.conf` file to enable daemon tracing for all daemons.

Example: Syntax to Enable Daemon Tracing for all Daemons

```
trace
all = on
endtrace
```

The system writes trace files for each daemon to the following default location:

```
/var/opt/SUNWsamfs/trace/<daemon-name>
```

Alternatively, trace files can be turned on individually for the `sam-archiverd`(1M), `sam-catserverd`(1M), `sam-fsd`(1M), `sam-ftpd`(1M), `sam-recycler`(1M), and `sam-stagerd`(1M) processes. Example: Syntax to Enable `sam-archiverd(1M)` Tracing enables daemon tracing for the archiver in `/var/opt/SUNWsamfs/trace/sam-archiverd`, sets the name of the archiver trace file to filename, and defines a list of optional trace events or elements to be included in the trace file as defined in option-list.

Example: Syntax to Enable ***`sam-archiverd`***(1M) Tracing

```
trace
sam-archiverd = on
sam-archiverd.file = <filename>
sam-archiverd.options = <option-list>
sam-archiverd.size = 10M
endtrace
```

Daemon trace files are not automatically rotated by default. As a result, they can become very large, and they might eventually fill the `/var` file system. You can enable automatic trace file rotation in the `defaults.conf` file by using the daemon-name`.size` parameter.

The `sam-fsd`(1M) daemon invokes the `trace_rotate.sh` script when a trace file reaches the specified size. The current trace file is renamed filename.1, the next newest is renamed _ filename_.2, and so on, for up to seven generations. Example: Syntax to Enable sam-archiverd (1M) Tracing specifies that the archiver trace file is to be rotated when its size reaches 10 megabytes.

For detailed information on the events that can be selected for inclusion in a trace file, see the defaults.conf(4) man page.

Enabling Device Logging

Sun StorageTek SAM systems write messages for archiving devices (automated libraries and tape drives) in log files stored in /var/opt/SUNWsamfs/devlog. This directory of files contains one log file for each device, and each of these files contains device-specific information. Each removable-media device has its own device log, which is named after its Equipment OrdinalNumber (eq) as defined in the mcf file. There is also a device log for the historian (Equipment Type hy) with a file name equal to the highest eq value defined in the mcf file incremented by one.

You can use the devlog keyword in the defaults.conf file to set up device logging using the following syntax:

```
devlog _eq_ [<option-list>]
```

If eq is set to all, the event flags specified in option-list are set for all devices.

For option-list, specify a space-separated list of devlog event options. If option-list is omitted, the default event options are err, retry, syserr, and date. For information on the list of possible event options, see the samset(1M) man page.

You can use the samset(1M) command to turn on device logging from the command line. Note that the device logs are not maintained by the system, so you must implement a policy at your site to ensure that the log files are routinely rolled over.

[Example: Device Log Output Example|#GFSJZ] shows sample device log output using the default output settings. It shows the first initialization of a 9840A tape drive. The drive is specified as Equipment OrdinalNumber 31 in the mcf file.

Example: Device Log Output Example

```
skeeball # cat mcf
#
# Equipment       Eq    Eq     Family  Device    Additional
# Identifier      ORD  Type     Set    State     Parameters
#-----------      ---  ----     ------ ------     ----------
samfs1            10    ms      samfs1   on
/dev/dsk/c1t2d0s0 11    md      samfs1   on      /dev/rdsk/c1t2d0s0
#
samfs2            20    ms      samfs2   on
/dev/dsk/c1t2d0s1 21    md      samfs2   on      /dev/rdsk/c1t2d0s1
#
#
# ---------- STK ACSLS Tape Library -----------
#
# Equipment               Eq     Eq     Family   Device  Additional
# Identifier              Ord    Type    Set     State   Parameters
#-----------              ---    ----    ------  ------  ----------
/etc/opt/SUNWsamfs/stk30   30    sk      stk30     on         -
/dev/rmt/0cbn              31    sg      stk30     on         -
/dev/rmt/1cbn              32    sg      stk30     on         -
skeeball #
skeeball # ls /var/opt/SUNWsamfs/devlog
30 31 32 33
skeeball # more /var/opt/SUNWsamfs/devlog/31
2003/06/11 11:33:31*0000 Initialized. tp
2003/06/11 11:33:31*1002 Device is STK , 9840
2003/06/11 11:33:31*1004 Rev 1.28
2003/06/11 11:33:31*1005 Known as STK 9840 Tape(sg)
2003/06/11 11:33:37 0000 Attached to process 691
2003/06/11 14:31:29 1006 Slot 0
2003/06/11 14:31:29 0000 cdb - 08 00 00 00 50 00
2003/06/11 14:31:29 0000       00 00 00 00 00 00
2003/06/11 14:31:29 0000 sense - f0 00 80 00 00 00 50 12 00 00
2003/06/11 14:31:29 0000         00 00 00 01 00 00 00 00 00 00
2003/06/11 14:31:30 0000 cdb - 08 00 00 00 50 00
2003/06/11 14:31:30 0000       00 00 00 00 00 00
2003/06/11 14:31:30 0000 sense - f0 00 80 00 00 00 50 12 00 00
2003/06/11 14:31:30 0000         00 00 00 01 00 00 00 00 00 00
2003/06/11 14:31:31 0000 cdb - 08 00 00 00 50 00
2003/06/11 14:31:31 0000       00 00 00 00 00 00
2003/06/11 14:31:31 0000 sense - f0 00 80 00 00 00 50 12 00 00
2003/06/11 14:31:31 0000         00 00 00 01 00 00 00 00 00 00
2003/06/11 14:31:31 3021 Writing labels
2003/06/11 14:31:32 1006 Slot 0
2003/06/11 14:31:32 3003 Label 700181
2003/06/11 14:31:31 blocksize = 262144
.
.
```

Example: Device Log Output Example shows how, about three hours after the 9840A device is initialized, a tape from slot 0 is loaded into the tape drive for archiving. The tape is checked three times for its volume serial name (VSN) label, and each time the system reports that the media is blank. After three checks, the system concludes that the tape is blank, labels it, and then reports the VSN label (700181), the date, the time, and the media block size.

Troubleshooting Utilities

Table: Troubleshooting Utilities lists the utilities that are helpful in diagnosing Sun StorageTek SAM configuration problems.

Table: Troubleshooting Utilities

Utility	Description
`sam-fsd`(1M)	Initializes the environment. Debugs basic configuration problems, particularly with new installations.
`samu`(1M)	Provides a comprehensive display that shows the status of Sun StorageTek SAM file systems and devices. Allows the operator to control file systems and removable media devices.
`sls`(1)	Consists of an extended version of the GNU `ls`(1M) command. The `-D` option displays extended Sun StorageTek SAM attributes.

samset(1M)	Sets parameters within the Sun StorageTek SAM environment.
samexplorer(1M)	Generates Sun StorageTek SAM diagnostic reports. For more information, see SAMreports.

For more information about these utilities, consult the relevant man pages and the Sun StorageTek SAM documentation, particularly Sun StorageTek QFS File System Configuration and Administration Guide and the Sun StorageTek Storage Archive Manager Archive Configuration and Administration Guide.

Troubleshooting SAM-QFS Manager

Contents

Troubleshooting SAM-QFS Manager

This page describes how to troubleshoot issues that might happen with the SAM-QFS Manager.

Log and Trace Files

The SAM-QFS Manager automatically enables logging when it is installed, but you must enable tracing manually. To enable tracing for SAM-QFS Manager, follow the instructions in SAMQFS:Tracing.

Log rotation is not supported for log or trace files.
The following table lists the files that the SAM-QFS Manager uses for logging and tracing.

Table – SAM-QFS Manager Log and Trace Files

Activity	File Location	Created by the User?
SAM-QFS Manager logging	/var/tmp/fsmgr.overall.log	Run the /opt/SUNWfsmgr/bin/fsmgr_report script to generate this report when you need to submit a Sun case or an escalation.
SAM-QFS Manager logging	/var/log/webconsole/console/fsmgr.log	No
Tomcat web console logging	/var/log/webconsole/console/console_debug_log	No
Tracing for SAM-QFS Manager and native code	/var/log/webconsole/console/fsmgr.trace_syslog	Yes

The following sections describe the log and trace files.

SAM-QFS Manager Logging

To generate an overall SAM-QFS Manager troubleshooting report, run the `/opt/SUNWfsmgr/bin/fsmgr_report` command to create the `/var/tmp/fsmgr.overall.log` file.

This log file contains general system information such as OS version, host name, and environment variables. It also contains package and version information for the software packages that support SAM-QFS Manager, such as Java™ and Tomcat. It includes configuration files that impact or are modified by SAM-QFS Manager.

The log file also contains data from the following SAM-QFS Manager log files:

- `/var/log/webconsole/console/console_debug_log`
- `/var/log/webconsole/console/fsmgr.log`
- `/var/log/webconsole/console/fsmgr.trace_syslog`

The SAM-QFS Manager creates the `fsmgr.log` log file when the application starts. It records information about operations that the user performs, and whether those operations were successful. Do not delete or modify this file; doing so will cause logging to stop. When the web server restarts, it erases the contents of this file and creates a new `fsmgr.log` file.

The SAM-QFS Manager uses an additional file, `/var/webconsole/fsmgr.log.lck`, to ensure that only one process at a time writes to the log file. Do not delete or modify this lock file.

Web Server Logging

The Sun Java Web Console Framework creates the `/var/log/webconsole/console/console_debug_log` file. It includes console-specific information such as environment variable settings that the console uses and a record of users logged in to the console.

If this file becomes too large, you can delete it. The system creates another instance of this file the next time the web server restarts.

Tracing

The SAM-QFS Manager trace file records the following information:

- Messages regarding whether operations were successful.
- Functions invoked with the application stack. These can be verbose.
- Messages that are important to developers for debugging purposes.

Tracing is not enabled by default.

How to Enable Tracing for SAM-QFS Manager and Native Code

The `syslog` daemon performs detailed tracing for SAM-QFS Manager and for native code. Use the following procedure to enable detailed tracing.

1. Use the `touch`(1) command to create the trace file.

   For example:

   ```
   # touch /var/log/webconsole/console/fsmgr.trace_syslog
   ```

2. Use `vi`(1) or another editor to add the following line to the `/etc/syslog.conf` file:

   ```
   local6.debug    /var/log/webconsole/console/fsmgr.trace_syslog
   ```

   Use a tab character to separate the two fields in this line.

3. Type the following command:

   ```
   # pkill -HUP syslogd
   ```

4. (Optional) Enable trace file rotation.

Trace files can become very large. Use `logadm(1M)` to manage the trace file for SAM-QFS Manager.

> **Note -**
> You cannot use the `log_rotate.sh` script to manage the SAM-QFS Manager trace file.

How to Enable Tracing or to Adjust the Tracing Level

1. Use the following command to enable tracing or to adjust the tracing level:

```
# /opt/SUNWfsmgr/bin/fsmgr trace <trace-level>
```

For trace-level, specify one of the values shown in table below.

trace-level Argument	Tracing Requested
`off`	Disables tracing.
1	Enables tracing for very important messages only. This includes severe errors that occur within the application.
2	Enables tracing for moderately important messages. This includes level 1 messages as well as debugging statements within the application that are useful for developers.
3	Enables tracing for all messages. This includes level 1 and level 2 messages as well as entry and exit points of functions within the application on the stack.

You can enable and disable tracing dynamically during run time by using the `fsmgr(1M)` command.

SAM-QFS Manager Error Messages

This section shows some of the messages you might see when using the SAM-QFS Manager software.

An unrecoverable error occurred during the page display. If the problem persists, please restart the web server.

Click the HOME button to return to the Server Selection page, which is the default page of the SAM-QFS Manager application.

If the system cannot display the Server Selection page, go to the web server and enter the following command to restart the it:

```
# /usr/sbin/smcwebserver restart
```

Contact your Sun Support representative if the problem persists.

HTTP 500 Internal server error

Go to the web server and run the following command to restart it:

```
# /usr/sbin/smcwebserver restart
```

Contact your Sun Support representative if the problem persists.

The page cannot be displayed.

Go to the web server and run the following command to restart it:

```
# /usr/sbin/smcwebserver restart
```

Contact your Sun Support representative if the problem persists.

Starting Java(TM) Web Console Version 3.1.Startup failed. See /var/log/webconsole/console/console_debug_log for detailed error information.

Examine the contents of the following file on the web server:

```
/var/log/webconsole/console/console_debug_log
```

If the log says the port (6789) is in use by some other process, issue the following commands:

```
# pkill -9 noaccess
# /usr/sbin/smcwebserver restart
```

Contact your Sun Support representative if the problem persists.

Failed to create the filesystem mount_samfs: fopen(mnttab) error: : Too many open files

The system generates this message if you are trying to create a file system with a large number of LUNs. To remedy this problem, follow these steps:

1. On the file system server, use the `ps`(1) and `grep`(1) commands to find the process ID for the `fsmgmtd` process.

   For example:

   ```
   # ps -ef | grep fsmgmtd
   ```

2. Use the `plimit`(1) command to increase the descriptors for the process.

   For example:

   ```
   # plimit -n 512 <process-id>
   ```

   For process-id, specify the process number.

3. Create the file system.

General Problems

Incompatible Software Release

The SAM-QFS Manager management station can collect or discover data from any server that has the corresponding or prior update of the following packages installed locally:

- SUNWsamfsr and SUNWsamfsu
- SUNWqfsr and SUNWqfsu

For example, the SAM-QFS Manager release 5.0 software was released at the same time as the Sun QFS and Sun SAM release 5 software. Therefore, the SAM-QFS Manager release 5.0 browser interface can discover data from a server that has any of the following installed locally:

- Release 4, update 6, SUNWsamfsr and SUNWsamfsu packages
- Release 4, update 6, SUNWqfsr and SUNWqfsu packages
- Release 5 SUNWsamfsr and SUNWsamfsu packages
- Release 5 SUNWqfsr and SUNWqfsu packages

If the server has an older release level of the packages installed, “Not Supported” is displayed in the Release column on the Servers page.

If the server has a newer release level of the packages installed, a Down icon is displayed on the Servers page.

In either of these situations, do the following to obtain compatible release levels:

1. On the management station and on the server, use the pkginfo(1M) -l (the letter ell) command to verify the package release on the system.
2. Depending on the results of the pkginfo(1M) -l command, do the following:
   - If the packages on the management station and server are not compatible, upgrade the system that has the older release.
   - If the packages on the management station and server are compatible, use the showrev(1M) -p command on both the management station and server to view the patch release levels.

```
showrev -p /usr/xpg4/bin/grep -E 'SUNWsam|SUNWqfs'
```

For more information about compatible software and patch release levels, see the README document.

One Device Displayed Multiple Times

If a virtual device is exported from a storage box on two different ports, the current server might view this device as two separate devices. If it does, the SAM-QFS Manager browser interface will display the device two times.

To address this situation, you must install multipathing software on the server and then click Refresh in the browser interface.

Inaccessible Server

If the SAM-QFS Manager software cannot communicate with a server, a Down icon is displayed next to the server in the browser interface. There are several reasons this might occur:

- The SUNWqfsr and SUNWqfsu packages or the SUNWsamfsr and SUNWsamfsu packages might not be installed on the server or they might be at incompatible release levels. Search for these packages on the server to verify that they are installed.
- The SAM-QFS Manager daemon (fsmgmtd) might not be running on the server. Do the following:
  1. On the server, use the fsmadm(1M) status command to check whether the daemon is running.
  2. If it is not, manually restart the daemon by typing the following:

```
/opt/SUNWsamfs/sbin/fsmadm restart
```

  You can configure the daemon to start automatically by using the fsmadm(1M) config -a command on the server.

- The rpcbind process might not be running on the server. Do the following:
  1. Log in to the server as root.
  2. Restart the rpcbind process by typing the following:

```
/usr/sbin/rpcbind &
```

- The management station might not be able to remotely administer the server. All hosts that are listed in /opt/SUNWsamfs/sbin/fsmadm on the server can remotely administer the server. To add the management station to the list, log in

to the server as `root` and use the `fsmadm`(1M) `add` management-station-name.domain-name command.
- The server is unavailable, it is not connected to the network, or the network itself is unavailable. Check the server's cabling and investigate any possible networking issues.

No Remaining Space for a VSN

If you receive a notification that there is no space available for archiving on a volume (VSN), do one of the following:

- Add more VSNs to the copies of the appropriate archive policy. The appropriate archive policy is the policy that is applied to the files being archived to that VSN.
- Recycle the expired archive copies on the VSN by running the recycler.

> Note -
> You can use the Monitoring Console to track VSN usage. One of the available reports lists archive copies with the highest usage.

Damaged or Stale Files in a Recovery Point

If you encounter either of the following situations, you must archive the specified files and then create another recovery point of the archiving file system:

- You review the recovery point log file and you find a message that says a file is damaged and cannot be restored with the specified recovery point.
- You receive an email notification that a stale files exists and that the file cannot be restored using the specified recovery point.

Existing Fault Conditions Not Displayed

If you are aware that a server is in a fault condition, but there is no indicator of this fault condition in the browser interface, you might need to restart the Solaris sysevent daemon (`syseventd`) on the server.

The following explanation applies only to servers with version 4, update 5, or earlier versions of the software.

The reason this might occur is that `/opt` might not be part of the root file system on the server. If this is the case, the Solaris sysevent daemon (`syseventd`) starts before `/opt` is available when the server is rebooted. Because `/opt` is unavailable to the daemon upon rebooting, the daemon cannot find some of the `SUNWsamfs` binaries that are required for the SAM-QFS Manager software to report recent fault conditions or to send Recovery Point Interrupted or File System Overflow notifications.

To address this issue, perform the following:

1. After rebooting the server, log in to the server as `root`.
2. Type the following at the command prompt:

   ```
   pkill -HUP syseventd
   ```

   The Solaris sysevent daemon is restarted. The daemon can now find the required `SUNWsamfs` binaries.

Remote Procedure Call Daemon Information

The following procedure can help you obtain troubleshooting information for the remote procedure call (RPC) daemon, `fsmgmtd`(1M).

How to Determine Whether the RPC Daemon Is Running

1. Log in to the Sun SAM server.
2. Become superuser.
3. Display status information for the SAM-QFS Manager daemon (`fsmgmtd`):

   Issue the following command:

   ```
   # /opt/SUNWsamfs/sbin/fsmadm status
   ```

If the daemon is not running, it does not display its status. Enter the following command to start the daemon:

```
# /opt/SUNWsamfs/sbin/fsmadm config -a
```

This command also enables the daemon to restart automatically if it dies.

Troubleshooting the Archiver

Contents

Troubleshooting the Archiver

This page provides troubleshooting information about the archiver.

Archiving Tools

Upon initial setup, the archiver might not perform the tasks as intended. Make sure that you are using the following tools to monitor the archiving activity of the system:

- The SAM-QFS Manager software - To display archiving activity, go to the Servers page and click the name of the server for which you want to display file system information. Click the System Administration node in the navigation tree, and then select Monitoring Console to display system information like active daemons, file system, library and drive, or archiving activities.

  For complete information on using SAM-QFS Manager to monitor jobs, see the SAM-QFS Manager online help file.

- **samu**(1M) utility's **a** display - This display shows archiver activity for each file system. It also displays archiver errors and warning messages, such as the following:

  ```
  Errors in archiver commands - no archiving will be done
  ```

  The samu(1M) utility's a display includes messages for each file system. It indicates when the archiver will scan the .inodes file again and the files currently being archived.

- Archive logs - You can define these logs in the archiver.cmd file, and you should monitor them regularly to ensure that files are archived to volumes. Archive logs can become excessively large and should be reduced regularly either manually or through a cron(1) job. Archive these log files for safekeeping, because the information in them enables data recovery.

- **sfind**(1) command - 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) command - Files are not considered for release unless a valid archive copy exists. The sls -D command displays inode information for a file, including copy information.

  > Note
  > Output from the sls -D command might show the word archdone on a file. This is not an indication that the file has an archive copy. It is only an indication that the file has been scanned by the archiver and that all the work associated with the archiver itself has been completed. An archive copy exists only when you can view the copy information displayed by the sls(1) command.

Occasionally, you might see messages indicating that the archiver either has run out of space on cartridges or has no cartridges. These messages are as follows:

- When the archiver has no cartridges assigned to an archive set:

```
No volumes available for Archive Set <setname>
```

- When the archiver has no space on the cartridges assigned to an archive set:

```
No space available on Archive Set <setname>
```

Why Files Are Not Archiving

Reasons your Sun StorageTek SAM environment might not be archiving files include the following:

- The `archiver.cmd` file has a syntax error. Run the `archiver -lv` command to identify the error, then correct the flagged lines.
- The `archiver.cmd` file has a `wait` directive in it. Either remove the `wait` directive or override it by using the `samu`(1M) utility's `:arrun` command.
- No volumes are available. You can view this from `archiver`(1M) `-lv` command output. Add more volumes as needed. You might have to export existing cartridges to free up slots in the automated library.
- The volumes for an archive set are full. You can export cartridges and replace them with new cartridges (make sure that the new cartridges are labeled), or you can recycle the cartridges. For more information on recycling, see the Sun StorageTek Storage Archive Manager Archive Configuration and Administration Guide.
- The VSN section of the `archiver.cmd` file does not list correct media. Check your regular expressions and VSN pools to ensure that they are correctly defined.
- There is not enough space to archive any file on the available volumes. If you have larger files and it appears that the volumes are nearly full, the cartridges might be as full as the SAM environment allows. If this is the case, add cartridges or recycle.
- The `archiver.cmd` file has the `no_archive` directive set for directories or file systems that contain large files.
- The `archive`(1) `-n` (archive never) command has been used to specify too many directories, and the files are never archived.
- Large files are busy. Thus, they never reach their archive age and are not archived.
- Hardware or configuration problems exist with the automated library.
- Network connection problems exist between client and server. Ensure that the client and the server have established communications.

Additional Archiver Diagnostics

In addition to examining the items on the previous list, you should check the following when troubleshooting the archiver:

- The `syslog` file (by default, `/var/adm/sam-log`). This file can contain archiver messages that indicate the source of a problem.
- Volume capacity. Ensure that all required volumes are available and have sufficient space on them for archiving.
- The trace files. If the archiver appears to cause excessive, unexplainable cartridge activity or appears to be doing nothing, turn on the trace facility and examine the trace file. For information on trace files, see the `defaults.conf`(4) man page.
- The `truss`(1) `-p` pid command. You can use this command on the archiver process (`sam-archiverd`) to identify the system call that is not responding. For more information on the `truss`(1) command, see the `truss`(1) man page.
- The `showqueue`(1M) command. This command displays the content of the archiver queue files and displays the progress of archiving. You can use it to observe the state of archiver requests that are being scheduled or archived. Any archive request that cannot be scheduled generates a message indicating the reason.

Troubleshooting the Configuration Files

Contents

Troubleshooting the Configuration Files

After the software packages have been installed, you need to tailor the configuration files to the site installation in order to bring the system into an operational state. Syntactical and typographical errors in these configuration files manifest themselves in unexpected behavior.

Configuration Files and Their Locations

Configuration File Purpose	Default Location
Master configuration file	`/etc/opt/SUNWsamfs/mcf`
`st` device file	`/kernel/drv/st.conf`
`samst`(7) device file	`/kernel/drv/samst.conf`
Device mapping	`/etc/opt/SUNWsamfs/inquiry.conf`
Default settings file	`/etc/opt/SUNWsamfs/defaults.conf`

The `/etc/opt/SUNWsamfs/mcf` File

The `mcf` file defines the file system devices and device family sets.

The `mcf` file is read when `sam-fsd`(1M) is started. It can be changed at any time, even while `sam-fsd` is running, but `sam-fsd`(1M) recognizes `mcf` file changes only when the daemon is restarted. The following example shows an `mcf` file for an archiving environment.

Example – Example **`mcf`** File Configured for Archiving

```
#
# SAM-QFS file system configuration example
#
# Equipment       Eq Eq Family Dev  Additional
# Identifier      Or Tp Set    St   Parameters
# --------------- -- -- ------ --- ----------
samfs1            10 ms samfs1
/dev/dsk/c1t1d0s6 11 md samfs1 on
/dev/dsk/c2t1d0s6 12 md samfs1 on
/dev/dsk/c3t1d0s6 13 md samfs1 on
/dev/dsk/c4t1d0s6 14 md samfs1 on
/dev/dsk/c5t1d0s6 15 md samfs1 on
#
samfs2            20 ms samfs2
/dev/dsk/c1t1d0s0 21 md samfs2 on
/dev/dsk/c1t0d0s1 22 md samfs2 on

#
/dev/samst/c1t2u0 200 rb hp30 on
/dev/samst/c1t5u0 201 od hp30 on
/dev/samst/c1t6u0 202 od hp30 on


/dev/samst/c4t500104F0009C2F6Fu0 300 sn L500 on
/dev/rmt/0cbn                    301 li L500 on
/dev/rmt/1cbn                    302 li L500 on
```

For information about the format of the `mcf` file in detail, see About the Master Configuration File.

The most common problems with the `mcf` file are syntactical and typographical errors. The `sam-fsd`(1M) command is a useful tool in debugging the `mcf` file. If `sam-fsd` encounters an error as it processes the `mcf` file, it writes error messages to the SAM-QFS log file (if configured). It also reports errors detected in the following other files, if present:

- `diskvols.conf`
- `samfs.cmd`
- `defaults.conf`

For a newly created or modified `mcf` file, run the `sam-fsd`(1M) command and check for error messages. If necessary, correct the `mcf` file and rerun the `sam-fsd` command to ensure that the errors have been corrected. Repeat this process until all errors have been eliminated. When the `mcf` file is error free, re initialize the `sam-fsd` daemon by sending it the `SIGHUP` command. The following example shows this process.

Example – Checking the **mcf** File

```
skeeball # sam-fsd
6: /dev/dsk/c1t2d0s0 10 md samfs1 on /dev/rdsk/c1t2d0s0
*** Error in line 6: Equipment Number 10 already in use
1 error in "/etc/opt/SUNWsamfs/mcf"
sam-fsd: Read mcf /etc/opt/SUNWsamfs/mcf failed.
skeeball #
skeeball # cat mcf
#
# Equipment          Eq   Eq  Family Device Additional
# Identifier         ORD Type Set    State  Parameters
#-----------         --- ---- ------ ------ ----------
samfs1               10   ms   samfs1  on
/dev/dsk/c1t2d0s0    10   md   samfs1  on
#
samfs2               20   ms   samfs2  on
/dev/dsk/c1t2d0s1    21   md   samfs2  on
#
#
# ---------- STK ACSLS Tape Library -----------
#
# Equipment              Eq  Eq   Family Device Additional
# Identifier             Ord Type Set    State  Parameters
#-----------             --- ---- ------ ------ ----------
/etc/opt/SUNWsamfs/stk30 30   sk   stk30   on
/dev/rmt/0cbn            31   sg   stk30   on
/dev/rmt/1cbn            32   sg   stk30   on

skeeball #
<correct error>

skeeball # sam-fsd
Trace file controls:
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
            cust err fatal misc proc date
            size 0 age 0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
            cust err fatal misc proc date
            size 0 age 0
sam-fsd /var/opt/SUNWsamfs/trace/sam-fsd
            cust err fatal misc proc date
            size 0 age 0
sam-ftpd /var/opt/SUNWsamfs/trace/sam-ftpd
            cust err fatal misc proc date
            size 0 age 0
sam-recycler /var/opt/SUNWsamfs/trace/sam-recycler
            cust err fatal misc proc date
            size 0 age 0
sam-sharefsd /var/opt/SUNWsamfs/trace/sam-sharefsd
            cust err fatal misc proc date
            size 0 age 0
sam-stagerd /var/opt/SUNWsamfs/trace/sam-stagerd
            cust err fatal misc proc date
            size 0 age 0
Would stop sam-archiverd()
Would stop sam-ftpd()
Would stop sam-stagealld()
Would stop sam-stagerd()
Would stop sam-amld()
skeeball #
skeeball # samd config
skeeball #
```

Enable the changes to the `mcf` file for a running system by running the `samd`(1M) command with its `config` option (as shown at the end of the example above) or by sending the `SIGHUP` signal to `sam-fsd`. The procedure for reinitializing `sam-fsd` to make it recognize `mcf` file modifications varies, depending on the nature of the changes implemented in the `mcf` file. For the procedures to be followed in specific circumstances, see Sun QFS File System Configuration and Administration Guide.

Verifying **mcf** Drive Order Matching

For direct attached libraries with more than a single drive, the order in which drive entries appear in the `mcf` file must match the order in which they are identified by the library controller. The drive that the library controller identifies as the first drive must be the first drive entry for that library in the `mcf` file, and so on. To check the drive order for a direct attached library, follow the instructions in Checking the Drive Order in Libraries.

Network attached libraries use different procedures from direct attached libraries, because the drive order for a network attached library is defined by the library control software. For example, for a network attached StorageTek library, the drive mapping in the ACSLS parameters file must match the drives as presented by the ACSLS interface. In this case, the procedure is similar to that for a library without a front panel, except that an additional check is necessary to ensure that the ACSLS parameters file mapping is correct.

The **/kernel/drv/st.conf** File

Some tape devices that are compatible with SAM-QFS software are not supported by default in the Solaris OS kernel. The file `/kernel/drv/st.conf` is the Solaris `st`(7D) tape driver configuration file for all supported tape drives. This file can be modified to enable operation of normally unsupported drives with a SAM-QFS system. Attempting to use any such device in the SAM-QFS environment without updating the `st.conf` file, or with an incorrectly modified file, causes the system to write messages such as the following to the device log file:

```
Aug  3  19:43:36 samfs2 scanner[242]: Tape device 92 is default
type. Update /kernel/drv/st.conf
```

If your configuration is to include devices not supported by the Solaris OS, consult the `/opt/SUNWsamfs/examples/st.conf_changes` file for instructions on how to modify the `st.conf` file.

For example, the IBM LTO drive is not supported by default in Solaris kernel. The following example shows the lines you need to add to the `st.conf` file in order to include IBM LTO drives in a SAM-QFS environment.

Example – Lines to be Added to **st.conf**

```
"IBM ULTRIUM-TD1", "IBM Ultrium", "CLASS_3580",
CLASS_3580 = 1,0x24,0,0x418679,2,0x00,0x01,0;
```

The `st.conf` file is read only when the `st` driver is loaded, so if the `/kernel/drv/st.conf` file is modified, perform one of the following actions in order to direct the system to recognize the changes:

- Use the `unload`(1M) and `modload`(1M) command to reload the driver.
- Reboot the system.

The **/kernel/drv/samst.conf** File

The `samst`(7) driver for SCSI media changers and optical drives is used for direct attached SCSI or Fibre Channel tape libraries and for magneto-optical drives and libraries.

As part of the installation process, the SAM-QFS software creates entries in the `/dev/samst` directory for all devices that were attached and recognized by the system before the `pkgadd`(1M) command was entered to begin the installation.

If you add devices after running the `pkgadd`(1M) command, you must use the `devfsadm`(1M) command, as follows, to create the appropriate device entries in `/dev/samst`:

```
# /usr/sbin/devfsadm -i samst
```

After the command is issued, verify that the device entries have been created in `/dev/samst`. If they have not, perform a reconfiguration reboot and attempt to create the entries again.

If the `/dev/samst` device is not present for the automated library controller, the `samst.conf` file might need to be updated. In general, Fibre Channel libraries, libraries with targets greater than 7, and libraries with logical unit numbers (LUNs) greater than 0 require the `samst.conf` file to be updated. To add support for such libraries, add a line similar to the following to the `/kernel/drv/samst.conf` file:

```
name="samst" parent="fp" lun=0 fc-port-wwn="500104f00041182b";
```

In the previous example line, 500104f00041182b is the World Wide Name (WWN) port number of the fibre-attached automated library. If you need to, you can obtain the WWN port number from the `cfgadm`(1M) command's output. The following example shows this command.

Example – Using **`cfgadm`**(1M) to obtain the WWN

```
# cfgadm -al
Ap_Id Type Receptacle Occupant Condition
c0 scsi-bus connected configured unknown
c0::dsk/c0t0d0 disk connected configured unknown
c0::dsk/c0t6d0 CD-ROM connected configured unknown
c1 scsi-bus connected configured unknown
c2 scsi-bus connected unconfigured unknown
c4 fc-fabric connected configured unknown c4::210000e08b0645c1 unknown connected unconfigured
unknown
.
.
c4::500104f00041182b med-changer connected configured unknown
c4::500104f00043abfc tape connected configured unknown
c4::500104f00045eeaf tape connected configured unknown
c4::5005076300416303 tape connected configured unknown
.
```

For network attached tape libraries such as a StorageTek library controlled by ACSLS, the `samst` driver is not used, and no `/dev/samst` device entries are created.

The **`/etc/opt/SUNWsamfs/inquiry.conf`** File

The `/etc/opt/SUNWsamfs/inquiry.conf` file defines vendor and product identification strings for recognized SCSI or fibre devices and matches these with SAM-QFS product strings. If you have devices that are not defined in `inquiry.conf`, you need to update the file with the appropriate device entries. This is not a common practice, because the great majority of devices are defined in the file. The following example shows a fragment of the `inquiry.conf` file.

Example – Part of the **`inquiry.conf`** File

```
"ATL", "ACL2640", "acl2640" # ACL 2640 tape library
"HP", "C1160A", "hpoplib" # HP optical library
"IBM" "03590", "ibm3590" # IBM3590 Tape
"MTNGATE" "V-48" "metd28" # metrum v-48 tape library
"OVERLAND", "LXB", "ex210" # Overland LXB2210 robot
"Quantum" "DLT2000", "dlt2000" # digital linear tape
"STK", "9490", "stk9490" # STK 9490 tape drive
"STK", "97", "stk97xx" # STK 9700 series SCSI
"STK", "SD-3" "stkd3" # STK D3 tape drive
```

If changes to this file are required, you must make them and then reinitialize your SAM-QFS software by issuing the following commands:

```
# samcmd aridle
# samcmd stidle

Make sure the tape drives are not longer active

# samcmd unload <eq>

Repeat the above command for each tape drive

Once all the tape drives are unload proceed with the following commands:
# samd stop
# samd config
```

If the system detects errors in the inquiry.conf file during reinitialization, it writes messages to the SAM-QFS log file. Check for error messages similar to those shown in the following example after making changes to inquiry.conf and reinitializing the SAM-QFS software.

Example – Messages Regarding **inquiry.conf** Problems

```
.
May 22 16:11:49 ultra1 samfs[15517]: Unknown device, eq 30
("/dev/samst/c0t2u0"), dtype (0x8)
May 22 16:11:49 ultra1 samfs[15517]: Vender/product OVERLAND LXB.
May 22 16:11:49 ultra1 samfs[15517]: Update /etc/opt/SUNWsamfs/inquiry.conf
(see inquiry.conf(4)).
May 22 16:11:49 ultra1 samfs[15517]: Device being offed eq 30.
.
```

The /etc/opt/SUNWsamfs/defaults.conf File

The defaults.conf configuration file allows you to establish certain default parameter values for a SAM-QFS environment. The system reads the defaults.conf file is when sam-fsd(1M) is started or reconfigured. It can be changed at any time while the sam-fsd(1M) daemon is running. The changes take effect when the sam-fsd(1M) daemon is restarted, or when it is sent the signal SIGHUP. Temporary changes to many values can be made using the samset(1M) command.

The sam-fsd(1M) command is also useful for debugging the defaults.conf file. If the sam-fsd(1M) daemon encounters an error as it processes the defaults.conf file, it writes error messages to the SAM-QFS log file.

For a newly created or modified defaults.conf file, run the sam-fsd(1M) command and check for error messages. If necessary, correct the file and rerun the sam-fsd(1M) command to ensure that the errors have been corrected. Repeat this process until all errors have been eliminated.

If you modify the defaults.conf file on a running system, you need to reinitialize it by restarting the sam-fsd(1M) daemon. You can use the samd(1M) command with its config option to restart sam-fsd(1M). See Sun QFS File System Configuration and Administration Guide for the procedures to be followed in specific circumstances.

Troubleshooting the Recycler

The most frequent problem encountered with the recycler occurs when the recycler is invoked and generates a message similar to the following:

```
Waiting for VSN mo:OPT000 to drain, it still has 123 active archive copies.
```

One of the following conditions can cause the recycler to generate this message:

- The archiver has failed to rearchive the archive copies on the volume.
- The archive copies referred to in the message are not files in the file system. Rather, they are metadata archive copies.

The first condition can exist for one of the following reasons:

- Files that need to be rearchived are marked no_archive.
- Files that need to be rearchived are in the no_archive archive set.
- Files cannot be archived because there are no available volume serial numbers (VSNs).
- The archiver.cmd file contains a wait directive.

To determine which condition is in effect, run the recycler with the -v option. As Example: Recycler Messages shows, this option displays the path names of the files associated with the archive copies in the recycler log file.

Example: Recycler Messages

```
Archive copy 2 of /sam/fast/testA resides on VSN LSDAT1
Archive copy 1 of /sam3/tmp/dir2/filex resides on VSN LSDAT1
Archive copy 1 of Cannot find pathname for file system /sam3
inum/gen 30/1 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gunk/tstfilA00 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gunk/tstfilF82 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gunk/tstfilV03 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gink/tstfilA06 resides on VSN LSDAT1
Archive copy 1 of /sam7/hgm/gink/tstfilA33 resides on VSN LSDAT1
Waiting for VSN dt:LSDAT1 to drain, it still has 8 active archive
copies.
```

In this example output, messages are displayed that contain seven path names with one message displaying "`Cannot find pathname...` text." This message will appear only after a system crash occurs that partially corrupts the `.inodes` file. Determine why the seven files cannot be rearchived, resolve the problem(s), and then rearchive the seven files. Note that only one archive copy is not associated with a file.

To solve the problem of finding the path name, run `samfsck`(1M) to reclaim orphan inodes. If you choose not to run `samfsck`(1M), or if you are unable to unmount the file system to run `samfsck`(1M), you can manually relabel the cartridge after verifying that the `recycler -v` output is clean of valid archive copies. However, because the recycler continues to encounter the invalid inode remaining in the `.inodes` file, the same problem might recur the next time the VSN is a recycle candidate.

Another recycler problem occurs when the recycler fails to select any VSNs for recycling. To determine why each VSN was rejected, you can run the recycler with the `-d` option. This displays information on how the recycler selects VSNs for recycling.

Troubleshooting the Releaser

Reasons that the releaser might not release a file include the following:

- Files can be released only after they are archived. There might not be an archive copy. For more information about this, see Why Files Are Not Archiving.
- The archiver requested that a file not be released. This can occur under the following conditions:
    - The archiver has just staged an offline file to make an additional copy.
    - The `-norelease` directive in the `archiver.cmd` file was set, and all the copies flagged `-norelease` have not been archived. The releaser summary output displays the total number of files with the `archnodrop` flag set.

      The file is set for partial release, and the file size is less than or equal to the partial size rounded up to the disk allocation unit (DAU) size (block size).
- The file changed residence in the last min-residence-age minutes.
- The `release -n` command has been used to prevent directories and files from being released.
- The `archiver.cmd` file has the `-release n` option set for too many directories and files.
- The releaser high-water mark or low-water mark is set too high, and automatic releasing occurs too late or stops too soon. Verify this in the `samu`(1M) utility's `m` display or with File System Manager, and lower this value.
- Large files are busy. They will never reach their archive age, never be archived, and never be released.

Using SAMreports

The SAMreport is collected information from a Sun StorageTek SAM environment and it's an important aid to diagnosing complex Sun StorageTek SAM problems. A SAMreport is needed by an engineer in the event of an escalation. There are two ways to generate a SAMreport:

- Using the `samexplorer` script (called `info.sh` in versions prior to 4U1). The `samexplorer` script writes the SAMreport to file `/tmp/SAMreport.`*`hostname`*`.`*`YYYYMMDD`*`.`*`HHMMZ`*`.tar.gz`.

- Using the SAM-QFS Manager. Go to the SAM Explorer section on the Metrics & Reports->System Details page and click Generate Report. You can specify where to save the SAMReport file.

The SAMreport includes the following information:

- Packaging, revision levels, and licensing information
- Configuration files (`mcf`, `archiver.cmd`, `recycler.cmd`, `inquiry.conf`, `defaults.conf`)
- Log files (`sam-log`, messages, `archiver.log`, `recycler.log`, `releaser.log`, trace files)
- Memory dump information

If log files are not routinely collected, an important source of diagnostic information is missing from the SAMreport. It is important to ensure that sites implement a comprehensive logging policy as part of their standard system administration procedures.

It is recommended that the SAMreport be generated in the following circumstances:

- Whenever there is a system panic, core dump, crash, hang, or stall
- As close to any system event as possible

Run the `samexplorer` script and save the SAMreport file before attempting recovery. Ensure that SAMreport is moved from `/tmp` before rebooting. By default, the `samexplorer` output is written to a single compressed `tar` archive containing all of the `samexplorer` output files. If you would like to generate the individual files in an uncompressed format, you can use the `samexplorer -u` option.

The functionality of `samexplorer` has been fully incorporated into the Sun Explorer Data Collector, release 4U0. However, `samexplorer` provides a focused set of data tuned to the Sun StorageTek SAM environment that can be quickly and simply collected and sent to escalation engineers for rapid diagnosis.

SAM-QFS Glossary

SAM-QFS Glossary

For storage industry standard definitions, see http://www.snia.org/education/dictionary/.

Term	Definition
addressable storage	The storage space encompassing online, nearline, offsite, and offline storage that is user-referenced through a Sun QFS file system.
archive media	The media to which an archive file is written. Archive media can be removable tape or magneto-optical cartridges in a library. In addition, archive media can be a mount point on another system.
archive set	An archive set identifies a group of files to be archived, and the files share common criteria that pertain to the size, ownership, group, or directory location. Archive sets can be defined across any group of file systems.
archive storage	Copies of file data that have been created on archive media.
archiver	The archive program that automatically controls the copying of files to removable cartridges.
audit (full)	The process of loading cartridges to verify their VSNs. For magneto-optical cartridges, the capacity and space information is determined and entered into the automated library's catalog. See also, VSN.
automated library	A robotically controlled device designed to automatically load and unload removable media cartridges without operator intervention. An automated library contains one or more drives and a transport mechanism that moves cartridges to and from the storage slots and the drives.
backup storage	A snapshot of a collection of files for the purpose of preventing inadvertent loss. A backup includes both the file's attributes and associated data.
block allocation map	A bitmap representing each available block of storage on a disk and indicating whether the block is in use or free.
block size	See DAU.
cartridge	A physical entity that contains media for recording data, such as a tape or optical disk. Sometimes referred to as a piece of media, a volume, or the medium.
catalog	A record of the VSNs in an automated library. There is one catalog for each automated library and, at a site, there is one historian for all automated libraries. See also, VSN.
client-server	The model of interaction in a distributed system in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called the client. The program satisfying the response is called the server.
connection	The path between two protocol modules that provides reliable stream delivery service. A TCP connection extends from a TCP module on one machine to a TCP module on the other.
data device	In a file system, a device or group of devices upon which file data is stored.

DAU	Disk allocation unit. The basic unit of online storage. Also called block size.
device logging	A configurable feature that provides device-specific error information used to analyze device problems.
device scanner	Software that periodically monitors the presence of all manually mounted removable devices and that detects the presence of mounted cartridges that can be requested by a user or other process.
direct access	A file attribute (stage never) designating that a nearline file can be accessed directly from the archive media and need not be retrieved to disk cache.
direct attached library	An automated library connected directly to a server using a SCSI interface. A SCSI-attached library is controlled directly by the Sun Storage Archive Manager software.
direct I/O	An attribute used for large block-aligned sequential I/O. The `setfa`(1) command's `-D` option is the direct I/O option. It sets the direct I/O attribute for a file or directory. If applied to a directory, the direct I/O attribute is inherited.
directory	A file data structure that points to other files and directories within the file system.
disk allocation unit	See DAU.
disk buffer	In a Sun SAM-Remote configuration, the buffer on the server system that is used for archiving data from the client to the server.
disk cache	The disk-resident portion of the file system software, used to create and manage data files between online disk cache and archive media. Individual disk partitions or an entire disk can be used as disk cache.
disk space threshold	The maximum or minimum level of disk cache utilization, as defined by an administrator. The releaser controls disk cache utilization based on these predefined disk space thresholds.
disk striping	The process of recording a file across several disks, thereby improving access performance and increasing overall storage capacity. See also striping.
drive	A mechanism for transferring data to and from a removable media volume.
Ethernet	A local-area, packet-switched network technology. Originally designed for coaxial cable, it is now found running over shielded, twisted-pair cable. Ethernet is a 10- or 100-Mbytes/second LAN.
extent array	The array within a file's inode that defines the disk location of each data block assigned to the file.
family device set	See family set.
family set	A storage device that is represented by a group of independent physical devices, such as a collection of disks or the drives within an automated library. See also storage family set.
FDDI	Fiber-distributed data interface (FDDI) is a standard for data transmission in a local area network that can extend in range up to 200 km (124 miles). The FDDI protocol is based on the token ring protocol.
Fibre Channel	The ANSI standard that specifies high-speed serial communication between devices. Fibre Channel is used as one of the bus architectures in SCSI-3.
file system	A hierarchical collection of files and directories.
file-system-specific directives	Archiver and releaser directives that follow global directives in the `archiver.cmd` file, are specific to a particular file system, and begin with `fs =`. File-system-specific directives apply until the next `fs =` directive line or the end of file is encountered. If multiple directives affect a file system, the file-system-specific directives override the global directives.
FTP	File transfer protocol. An Internet protocol for transferring files between two hosts over a TCP/IP network.
global directives	Archiver and releaser directives that apply to all file systems and that appear before the first `fs =` line.
grace period	For disk quotas, the amount of time for which a user is allowed to create files and allocate storage after reaching the soft limit.
hard limit	For disk quotas, the maximum limit on file system resources, blocks, and inodes that users cannot exceed.
hosts file	The hosts file contains a list of all of the hosts in a shared file system. If you are initializing a file system as a Sun QFS shared file system, the hosts file, `/etc/opt/SUNWsamfs/hosts.`fs-name, must be created before the file system is created. The sammkfs(1M) command uses the hosts file when it creates the file system. You can use the samsharefs(1M) command to replace or update the contents of the hosts file at a later date.

indirect block	A disk block that contains a list of storage blocks. File systems have up to three levels of indirect blocks. A first-level indirect block contains a list of blocks used for data storage. A second-level indirect block contains a list of first-level indirect blocks. A third-level indirect block contains a list of second-level indirect blocks.
inode	Index node. A data structure used by the file system to describe a file. An inode describes all the attributes associated with a file other than the name. The attributes include ownership, access, permission, size, and the file location on the disk system.
inode file	A special file (`.inodes`) on the file system that contains the inode structures for all files resident in the file system. Inodes are 512 bytes long. The inode file is a metadata file, which is separated from file data in the file system.
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.
lease	A function that grants a client host permission to perform an operation on a file for a specified period of time. The metadata server issues leases to each client host. The leases are renewed as necessary to permit continued file operations.
library	See automated library.
library catalog	See catalog.
local file system	A file system that is installed on one node of a Sun Cluster system and is not made highly available to another node. Also, a file system that is installed on a server.
LUN	Logical unit number.
`mcf`	Master configuration file. The file that is read at initialization time that defines the relationships between the devices (the topology) in a file system environment.
media	Tape or optical disk cartridges.
media recycling	The process of recycling or reusing archive media with few active files.
metadata	Data about data. Metadata is the index information used to locate the exact data position of a file on a disk. It consists of information about files, directories, access control lists, symbolic links, removable media, segmented files, and the indexes of segmented files.
metadata device	A device (for example, a solid-state disk or mirrored device) upon which file system metadata is stored. Having file data and metadata on separate devices can increase performance. In the `mcf` file, a metadata device is declared as an `mm` device within an `ma` file system.
mirror writing	The process of maintaining two copies of a file on disjointed sets of disks to prevent loss from a single disk failure.
mount point	The directory on which a file system is mounted.
multireader file system	A single-writer, multireader capability that enables you to specify a file system that can be mounted on multiple hosts. Multiple hosts can read the file system, but only one host can write to the file system. Multiple readers are specified with the `-o reader` option with the `mount`(1M) command. The single-writer host is specified with the `-o writer` option with the `mount`(1M) command. For more information on the `mount`(1M) command, see the `mount_samfs`(1M) man page.
name space	The metadata portion of a collection of files that identifies the file, its attributes, and its storage locations.
nearline storage	Removable media storage that requires robotic mounting before it can be accessed. Nearline storage is usually less expensive than online storage, but it takes somewhat longer to access.
network attached automated library	A library, such as those from StorageTek, ADIC/Grau, IBM, or Sony, that is controlled using a software package supplied by the vendor. The QFS file system interfaces with the vendor software using a Sun Storage Archive Manager media changer daemon designed specifically for the automated library.
NFS	Network file system. A file system distributed by Sun that provides transparent access to remote file systems on heterogeneous networks.
NIS	The Sun OS 4.0 (minimum) Network Information Service. A distributed network database containing key information about systems and users on the network. The NIS database is stored on the master server and all slave servers.
offline storage	Storage that requires operator intervention for loading.

offsite storage	Storage that is remote from the server and is used for disaster recovery.
online storage	Storage that is immediately available, such as disk cache storage.
partition	A portion of a device or a side of a magneto-optical cartridge.
preallocation	The process of reserving a contiguous amount of space on the disk cache for writing a file. Preallocation can be specified only for a file that is size zero. For more information, see the `setfa`(1) man page.
pseudo device	A software subsystem or driver with no associated hardware.
QFS	The file system features of SAM-QFS. A QFS file system can be used separately from the Sun Storage Archive Manager archiving features.
quota	The amount of system resources that a user is allowed to consume.
RAID	Redundant array of independent disks. A disk technology that uses several independent disks to reliably store files. It can protect against data loss from a single disk failure, can provide a fault-tolerant disk environment, and can provide higher throughput than individual disks.
recycler	A Sun Storage Archive Manager utility that reclaims space on cartridges that is occupied by expired archive copies.
release priority	The priority according to which a file in a file system is released after being archived. Release priority is calculated by multiplication of various weights of file properties and then summation of the results.
releaser	A Sun Storage Archive Manager component that identifies archived files and releases their disk cache copies, thus making more disk cache space available. The releaser automatically regulates the amount of online disk storage according to high and low thresholds.
remote procedure call	See RPC.
removable media file	A special type of user file that can be accessed directly from where it resides on a removable media cartridge, such as magnetic tape or optical disk cartridge. Also used for writing archive and stage file data.
robot	The portion of an automated library that moves cartridges between storage slots and drives. Also called a transport.
round-robin	A data access method in which entire files are written to logical disks in a sequential fashion. When a single file is written to disk, the entire file is written to the first logical disk. The second file is written to the next logical disk, and so on. The size of each file determines the size of the I/O. See also disk striping and striping.
RPC	Remote procedure call. The underlying data exchange mechanism used by NFS to implement custom network data servers.
SAM	See SAM-QFS.
SAM-QFS	A configuration that combines the Sun Storage Archive Manager software with the Sun QFS file system. This configuration enables you to archive file system data by combining a high-speed, standard UNIX file system interface with the storage and archive management utilities. Previously marketed with the SAM-FS file system, beginning with release 5.0, SAM-QFS includes the QFS file system. SAM-QFS uses the SAM-QFS command set as well as standard UNIX file system commands.
`samfsdump`	A program that creates a control structure dump and copies all the control structure information for a given group of files. It is analogous to the UNIX `tar`(1) utility, but it does not generally copy file data. See also samfsrestore.
`samfsrestore`	A program that restores inode and directory information from a control structure dump. See also samfsdump.
SCSI	Small Computer System Interface. An electrical communication specification commonly used for peripheral devices such as disk and tape drives and automated libraries.
shared hosts file	When you create a shared file system, the system copies information from the hosts file to the shared hosts file on the metadata server. You update this information when you issue the samsharefs(1M) -u command
Small Computer System Interface	See SCSI.
soft limit	For disk quotas, a threshold limit on file system resources (blocks and inodes) that you can temporarily exceed. Exceeding the soft limit starts a timer. When you exceed the soft limit for the specified time, no further system resources can be allocated until you reduce file system use to a level below the soft limit.

staging	The process of copying a nearline or offline file from archive storage back to online storage.
storage family set	A set of disks that are collectively represented by a single disk family device.
storage slots	Locations inside an automated library in which cartridges are stored when not being used in a drive.
stripe size	The number of disk allocation units (DAUs) to be allocated before writing proceeds to the next device of a stripe. If the `stripe=0` mount option is used, the file system uses round-robin access, not striped access.
striped group	A collection of devices within a file system that is defined in the `mcf` file as one or more gXXX devices. Striped groups are treated as one logical device and are always striped with a size equal to the disk allocation unit (DAU).
striping	A data access method in which files are simultaneously written to logical disks in an interlaced fashion. SAM-QFS file systems provide two types of striping: “hard striping,” using stripe groups, and “soft striping,” using the `stripe=x` mount parameter. Hard striping is enabled when a file system is set up, and requires the definition of stripe groups within the `mcf` file. Soft striping is enabled through the `stripe=x` mount parameter, and can be changed for the file system or for individual files. It is disabled by setting `stripe=0`. Hard and soft striping can both be used if a file system is composed of multiple stripe groups with the same number of elements. See also round-robin.
Sun SAM-Remote client	A Sun Storage Archive Manager system with a client daemon that contains a number of pseudodevices, and can also have its own library devices. The client depends on a Sun SAM-Remote server for archive media for one or more archive copies.
Sun SAM-Remote server	Both a full-capacity Sun Storage Archive Manager storage management server and a Sun SAM-Remote server daemon that defines libraries to be shared among Sun SAM-Remote clients.
superblock	A data structure in the file system that defines the basic parameters of the file system. The superblock is written to all partitions in the storage family set and identifies the partition’s membership in the set.
tar	Tape archive. A standard file and data recording format used for archive images.
TCP/IP	Transmission Control Protocol/Internet Protocol. The internet protocols responsible for host-to-host addressing and routing, packet delivery (IP), and reliable delivery of data between application points (TCP).
timer	Quota software that keeps track of the period starting when a user reaches a soft limit and ending when the hard limit is imposed on the user.
`vfstab` file	The `vfstab` file contains mount options for the file system. Mount options specified on the command line override those specified in the `/etc/vfstab` file, but mount options specified in the `/etc/vfstab` file override those specified in the `samfs.cmd` file.
volume	A named area on a cartridge for sharing data. A cartridge has one or more volumes. Double-sided cartridges have two volumes, one on each side.
volume overflow	A capability that enables the system to span a single file over multiple volumes. Volume overflow is useful for sites using very large files that exceed the capacity of their individual cartridges.
VSN	Volume serial name. In the context of archiving to removable media cartridges, the VSN is a logical identifier for magnetic tape and optical disk that is written in the volume label. In the context of archiving to disk cache, this is the unique name for the disk archive set.
WORM	Write once read many. A storage classification for media that can be written only once but read many times.

Best Practices

Best Practices for SAM and QFS

Use this page to gather best practices information. It doesn't have to pretty, just useful.

- Best Practice - Installing and Configuring SAM-QFS
- Best Practices for disk archiving

Best Practice - Installing and Configuring SAM-QFS

Best Practice - Installing and Configuring SAM-QFS

Contributed by Timothy Whalen, Sun Microsystems
On January 7, 2009

The information below is based on the hardware purchased. We have found that the following applies to 95% of our customer base (regardless of use). Most customers have the following:

- One Tape library
- One Media type
- Ten or less tape drives
    - Most sites have four

===

Each new install/upgrade needs to have Solaris 10 update 6 installed. SAM-QFS 5.X will only be supported in Solaris 10 update 6 or newer releases

===

Based on the above we then match SAM-QFS to work with the hardware. The archiver.cmd file example below should be used at most SAM-QFS sites.

Global parameters:

```
archivemeta = off
examine = noscan
```

If no disk archiving is in place then use the following settings:

```
params
allsets -sort path -offline_copy stageahead  -reserve set
allsets.1  -startage 8h  -startsize 8G  -drives X -archmax 10G
allsets.2  -startage 24h -startsize 20G -drives X -archmax 24G
endparams
```

-
    - If a customer can not wait potential 8 hours before the file is written to tape then they should implement disk archiving.

If disk archiving is in place then use the following settings:

```
params
allsets -sort path -offline_copy stageahead
allsets.1  -startage 10m -startsize 500M  -drives 10 -archmax 1G
allsets.2  -startage 24h -startsize 20G   -drives X  -archmax 24G -reserve set
allsets.3  -startage 48h -startsize 20G   -drives X  -archmax 24G -reserve set
endparams
```

The releaser.cmd file should be change to:

```
list_size = 300000
```

The stager.cmd file

```
maxactive  =  500000  # If server has more than 8G of RAM
maxactive  =  100000  # If server has less than 8G of RAM
```

For most customers they should be using the "ms" file system configuration. In addition the segment size set on the disk storage should be 512K. The storage should be configured with RAID 5 3+1 or 4+1 (no virtual volumes).

```
samfs1                                         20 ms samfs1 on
/dev/dsk/c6t600A0B80002A0C2E000008DD493FEAFDd0s6 21 md samfs1 on
/dev/dsk/c6t600A0B80002A0C2E000008DF493FEB6Dd0s6 22 md samfs1 on
/dev/dsk/c6t600A0B80002A1F5C000009344940690Ad0s6 23 md samfs1 on
/dev/dsk/c6t600A0B80002A1F5C000009364940697Ed0s6 24 md samfs1 on
```

The more dedicated data LUNs and HBAs the faster the "ms" file system will perform. The number of server I/O slots have an impact on this recommendation.

If the above is followed customers can have up to 200 million files in a single file system. A samfsrstore should take less than 24 hours for 200 million files. This assumes the server has enough CPU and memory to support the environment.

We find customers who use the above configurations have less support calls. This includes less tape hardware problems, and tape media problems. Also by using the above settings the tape drives can be used more for staging files. These settings reduces the issue of the archiver using the tape drives all the time so the stager can not use them.

Sites the select the "ma" file system configuration should do so based on I/O requirements. When using the "ma" file system configuration there needs to be at least on dedicated "mm" device. This device is used to storage metadata (inode) information.

The "mm" device should be hardware RAID 10. This will require four dedicated fibre disk drives for each "mm" device.

```
images                                         30 ma images on
/dev/dsk/c6t600A0B80002A0C2E000008DD493FEAFDd0s1 31 mm images on
/dev/dsk/c6t600A0B80002A0C2E000008DF493FEB6Dd0s6 32 mr images on
/dev/dsk/c6t600A0B80002A1F5C000009344940690Ad0s6 33 mr images on
/dev/dsk/c6t600A0B80002A1F5C000009364940697Ed0s6 34 mr images on
/dev/dsk/c6t600A0B80002A1F5C0000093649406DBEd0s6 35 mr images on
```

Best Practices for disk archiving

Best Practices for disk archiving
Contributed by Tim Whalen
7/02/2009

==
Disk Archiving - Recommendations/Configurations
==
Having multiple dedicated disk archive file systems is the best approach. Having multiple dedicated disk archive file system yields the best performance for archiving and staging. We suggest the creation of one main directory for access to all the disk archives, for example:
/darc

Below the above directory are mount points for each disk archive file system:
/darc/DISK01
/darc/DISK02
/darc/DISK03

*Do not create multiple subdirectories in a disk archive file system in order to use each
of those subdirectories as a disk archive!* This will cause SAM-QFS to unnecessarily compete
for resources on the same disk.

The /etc/opt/SUNWsamfs/diskvols.conf file should look something like:
DISK01 /darc/DISK01
DISK02 /darc/DISK02
DISK03 /darc/DISK03

The disk archive file system can be UFS, ZFS, or SAM-QFS file system. If using a SAM-QFS file system for the disk archive, the /etc/opt/SUNWsamfs/mcf file could look like this:

```
disk01                                              20 ms disk01 on
/dev/dsk/c6t600A0B80002A0C2E000008DD493FEAFDd0s6  21 md disk01 on
/dev/dsk/c6t600A0B80002A0C2E000008DF493FEB6Dd0s6  22 md disk01 on
/dev/dsk/c6t600A0B80002A1F5C000009344940690Ad0s6  23 md disk01 on
/dev/dsk/c6t600A0B80002A1F5C000009364940697Ed0s6  24 md disk01 on
```

If the disk archive file system is SAM-QFS the /etc/vfstab entry should look like:

```
disk01   -   /darc/DISK01  samfs  -   yes   nosam,stripe=2
```

- - The above assumes the disk archive file system is not archiving to tape.

Note -
Keep the number of disk archive file systems between 15 to 30. Try to keep each disk archive file system between 10TB to 20TB in size (disk space providing). If a site has 8TB of disk space for disk archiving the site should create just one 8TB disk archive file system.