Oracle® Hierarchical Storage Manager and StorageTek QFS Software

File System Recovery Guide

Release 6.1.1

E42065-05

February 2017

Oracle Hierarchical Storage Manager and StorageTek QFS Software File System Recovery Guide, Release 6.1.1

E42065-05

Primary Author: Robert Craig Johnson

Contents

5 Recovering Lost and Damaged Files

6 Finishing Up

A Understanding Archiver and Migration Logs

B Glossary of Equipment Types

C Product Accessibility Features

Glossary

Preface

This document addresses the needs of system administrators, storage and network administrators, and service engineers who may have to recover damaged or lost files, file systems, and archiving solutions using Oracle Hierarchical Storage Manager (formerly StorageTek Storage Archive Manager).

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Prerequisites for Using this Document

This document assumes that you are already familiar with Oracle Solaris operating system, storage, and network administration. Please refer to the Solaris documentation and man pages and to storage hardware documentation for information on relevant tasks, commands, and procedures.

Conventions

The following textual conventions are used in this document:

- *Italic* type represents book titles and emphasis.
- `Monospace` type represents commands and text displayed in a terminal window and the contents of configuration files, shell scripts, and source code files.
- **`Monospace bold`** type represents user inputs and significant changes to commandline output, terminal displays, or file contents. It may also be used to emphasize particularly relevant parts of a file or display.
- ***`Monospace bold oblique`*** type represents variable inputs and outputs in a terminal display or file.
- *`Monospace oblique`* type represents other variables in a terminal display or file.

- **...** (three-dot ellipsis marks) represent file contents or command output that is not relevant to the example and has thus been omitted for brevity or clarity.
- `[ - ]` (brackets surrounding values separated by a hyphen) delimit value ranges.
- `[ ]` (brackets) in command syntax descriptions indicate optional parameters.
- `root@`*`solaris-host`*`:~#` represents a Solaris command shell prompt.
- `[root@`*`linux-host`*` ~]#` represents a Linux command shell prompt.

Available Documentation

The *Oracle Hierarchical Storage Manager and StorageTek QFS Software File System Recovery Guide* is part of the multivolume *Oracle HSM Customer Documentation Library*, available from `http://docs.oracle.com/en/storage/#sw`.

Oracle Solaris operating system documentation is available at `http://docs.oracle.com/en/operating-systems/`.

1

Introduction

This document outlines the steps that you should take in order to recover Oracle Hierarchical Storage Manager and StorageTek QFS Software, files, and file systems that have been lost or corrupted due to hardware failure, misconfiguration, human error, or the physical destruction of facilities and equipment. Properly configured Oracle HSM file systems are extremely robust. But the steps that you need to take during recovery—and your probability of success—depend on your degree of preparedness. So this introduction begins with an overview of the recovery process. Then it moves to a review of the data- and file system-protection measures that Oracle recommends. Finally it outlines the recovery steps that are open to you, given the preparations that you have made and the resources that you currently have available.

Failure and Recovery Scenarios

The scope of a file-system failure and the nature of the required recovery actions depend on the nature of the underlying problem. For example:

- If the server host fails, the Oracle HSM software and file-system configurations may be lost, leaving file system data and metadata intact but inaccessible until the configuration information is restored.

  Once the underlying hardware problem has been addressed and the operating system has been restored, you reinstall the software and restore the configuration files from backup copies. In this situation, follow the procedures outlined in Chapter 3, "Restoring the Oracle HSM Configuration".

- If an administrator inadvertently deletes or corrupts one or more configuration files, library catalogs, scripts, or `crontab` entries, access to one or more file systems may be lost along with some or all software functionality.

  You restore the configuration files from backup copies. Follow the procedures outlined in Chapter 3, "Restoring the Oracle HSM Configuration".

- If a disk or RAID group that provides the disk cache for the data in a standalone (non-archiving) QFS file-system fails, all files in the disk cache are lost.

  Once the hardware problem has been addressed, you restore lost files from QFS backup copies. See "Recovering Files Using a Recovery Point File" on page 5-1.

- If a disk or RAID group that provides the disk cache for the data in an archiving file-system fails, all files in the disk cache are lost.

  Once the hardware problem has been addressed, you restore files from archived copies or from Oracle HSM backup files. See "Recovering Files Using a Recovery Point File" on page 5-1 and "Recovering Files Using Log Entries" on page 5-3.

- If the disks that store file system metadata fail, the file system is lost and the data is no longer readily accessible.

  Once the hardware problem has been addressed, you restore metadata from backup files. If the metadata for an archiving file system was not backed up, it can be reconstructed from backup copies of the archiver log file and media-migration log files (if any). See Chapter 5, "Recovering Lost and Damaged Files".

- If an administrator inadvertently formats the disk partitions that host an Oracle HSM file system or issues the `sammkfs` command against existing Oracle HSM partitions, all files and metadata are lost.

  You restore metadata from backup files or reconstruct it from the archiver log and media-migration log files (if any) of an archiving file system. Data can be restored from archival media or from a backup file. See Chapter 5, "Recovering Lost and Damaged Files".

Recommended Preparations

In the *Oracle Hierarchical Storage Manager and StorageTek QFS Installation and Configuration Guide*, Oracle recommends that you take the following configuration, file-system, and data backup steps during your initial configuration:

- Store critical data in Oracle HSM archiving file systems.

  Archive at least two copies of the file data. Archive at least one copy on removable media, such as magnetic tape.

  If possible, configure disk archives on independent file systems that do not share physical devices with the disk cache of the archiving file-system.

- Store file-system metadata on highly redundant, mirrored storage.

- Regularly backup Oracle HSM file systems with recovery point files.

  A recovery point file stores file-system metadata and, optionally, data, so that files or entire file systems can be restored.

  If you have the Oracle Hierarchical Storage Manager software installed, you create recovery point files by running the `samfsdump` command. If you have only the QFS file system software, you use the `qfsdump` command. You can run the dump commands from the command line or from the Oracle HSM Manager graphical user interface.

  Using either command on its own backs up the metadata. Using either command with the `-U` option backs up data as well as metadata. The `-U` option is mainly useful for protecting file systems that are not archived to removable media.

- Configure the host to automatically save Oracle HSM metadata recovery point files. Create entries in the Solaris `crontab` file, or use the scheduling feature of the Oracle HSM Manager.

- Configure the host to automatically save the Oracle HSM archiver log file and media-migration log files (if any). Create entries in the Solaris `crontab` file.

  For each file that is archived or migrated to new media by the Oracle Hierarchical Storage Manager software, log files record the file's name and location (path) within the file system, the name of the archive (`tar`) file that holds copies, the removable media volumes that holds the archive file, and the position of the archive file on the media.

- Save backup copies of configuration files, `crontab` entries, and custom file-system management scripts (if any).
- Select a secure storage location for the Oracle HSM recovery information.

  Select an independent file system that you can mount on the Oracle HSM file system host.

  Make sure that the selected file system does not share any physical devices, logical volumes, partitions, or LUNs with the archiving file system. Do not store disaster recovery resources in the file system that they are meant to protect.

2

Stabilizing the Situation

Whenever you are faced with recovering from a significant file-system failure or potential data loss, your first steps should stabilize the affected systems, minimize chances for further losses, and preserve diagnostic information, where possible. This chapter outlines the actions that you need to take:

- stopping archiving and recycling processes (if any)
- preserving unarchived data
- preserving configuration and state information.

Stopping Archiving and Recycling Processes

When you have to restore an archiving file system or significant numbers of lost files, you should first stop the archiving and recycling processes for the file system. You want to stabilize and isolate the archive until you have assessed the situation and, ideally, restored everything to normal. Otherwise, ongoing archiving and recycling operations can, in some situations, make matters worse. Archiving and staging processes may propagate corrupted files. Recycling processes may delete the only remaining copies of valid data.

So, whenever possible, take the precautions listed below:

- stop archiving
- stop recycling.

Once recovery operations are complete, you can reverse the changes below and restore normal file system behavior.

Stop Archiving

1. Log in to the file-system metadata server as `root`.

   ```
   root@mds1:~#
   ```

2. Open the `/etc/opt/SUNWsamfs/archiver.cmd` file in a text editor, and scroll down to the first `fs` (file-system) directive.

   In the example, we use the `vi` editor:

   ```
   root@mds1:~# vi /etc/opt/SUNWsamfs/archiver.cmd
   # Configuration file for Oracle HSM archiving file systems
   #-----------------------------------------------------------------------
   # General Directives
   archivemeta = off
   examine = noscan
   ```

```
#-----------------------------------------------------------------------
# Archive Set Assignments
fs = hsmfs1
logfile = /var/adm/hsmfs1.archive.log
all .
    1 -norelease 15m
    2 -norelease 15m
fs = hsmfs2
logfile = /var/adm/hsmfs2.archive.log
all .
...
```

3. If you need to stop archiving on all file systems, insert a `wait` directive just before the first `fs` directive in the `archiver.cmd`. Save the `archiver.cmd` file, and close the editor.

   In the example, we insert the `wait` directive just before the directive for the `hsmfs1` file system, where it will apply to all file systems configured for archiving:

```
root@mds1:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#-----------------------------------------------------------------------
# Archive Set Assignments
wait
fs = hsmfs1
logfile = /var/adm/hsmfs1.archive.log
all .
    1 -norelease 15m
    2 -norelease 15m
    3 -norelease 15m
fs = hsmfs2
...
:wq
root@mds1:~#
```

4. If you need to stop archiving on only one file system, insert a `wait` directive just after the `fs` directive for that file system. Save the `archiver.cmd` file, and close the editor.

   In the example, we stop archiving activity on the `hsmfs1` file system:

```
root@mds1:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#-----------------------------------------------------------------------
# Archive Set Assignments
fs = hsmfs1
wait
logfile = /var/adm/hsmfs1.archive.log
all .
    1 -norelease 15m
    2 -norelease 15m
    3 -norelease 15m
fs = hsmfs2
...
:wq
root@mds1:~#
```

5. Next, stop recycling.

Stop Recycling

1. Log in to the file-system metadata server as `root`.

   ```
   root@mds1:~#
   ```

2. Open the `/etc/opt/SUNWsamfs/recycler.cmd` file in a text editor.

   In the example, we use the `vi` editor:

   ```
   root@mds1:~# vi /etc/opt/SUNWsamfs/recycler.cmd
   # Configuration file for Oracle HSM archiving file systems
   #-----------------------------------------------------------------------
   logfile = /var/adm/recycler.log
   no_recycle tp VOL[0-9][2-9][0-9]
   lib1 -hwm 95 -mingain 60
   ```

3. Add the `-ignore` parameter to each recycling directive in the `recycler.cmd` file. Then save the file, and close the editor.

   The `recycler.cmd` file does not contain recycling directives unless you have configured recycling by library, rather than by archive sets. But check it now.

   In the example, we have one recycling directive for tape library `lib1`:

   ```
   root@mds1:~# vi /etc/opt/SUNWsamfs/recycler.cmd
   # Configuration file for Oracle HSM archiving file systems
   #-----------------------------------------------------------------------
   logfile = /var/adm/recycler.log
   no_recycle tp VOL[0-9][2-9][0-9]
   lib1 -hwm 95 -mingain 60 -ignore
   :wq
   root@mds1:~#
   ```

4. If you are recovering from loss or damage to one more archiving file systems, back up unarchived files before proceeding.

5. If you are recovering from a server problem or from loss or damage to file systems, save the Oracle HSM configuration before proceeding.

6. If you need to restore directories and files, decide whether you need to save the Oracle HSM configuration or go directly to Chapter 5, "Recovering Lost and Damaged Files".

Preserving Unarchived Data

Unarchived files may remain in the disk cache of a damaged archiving file system. No copies of these files exist in the archive. So, if you can, back them up to a recovery point file now. Proceed as follows:

Back Up Unarchived Files

1. Log in to the file-system metadata server as `root`.

   ```
   root@mds1:~#
   ```

2. Select a safe storage location for the recovery point.

   In the example, we create a subdirectory, `unarchived/`, under a directory that we created for recovery points during initial configuration. The `/zfs` file system has no devices in common with `/hsmfs1`, the file system that we are recovering:

   ```
   root@mds1:~# mkdir /zfs1/hsmfs_recovery/unarchived/
   ```

```
root@mds1:~#
```

3. Change to the file system's root directory.

   In the example, we change to the mount-point directory /hsmfs1:

```
root@mds1:~# cd /hsmfs1
root@mds1:~#
```

4. Backup any unarchived files that remain in the disk cache. Use the command samfsdump -u -f *recovery-point*, where *recovery-point* is the path and file name of the output file.

   The -u option causes the samfsdump command to back up any data files that have not been archived. In the example, we save the recovery point file 20160325 to the remote directory /zfs1/hsmfs_recovery/unarchived/:

```
root@mds1:~# samfsdump -u -f /zfs1/hsmfs_recovery/unarchived/20160325
root@mds1:~#
```

5. If you are recovering from a server problem or from loss or damage to file systems, save the Oracle HSM configuration before proceeding.

6. If you need to restore directories and files, decide whether you need to save the Oracle HSM configuration or go directly to Chapter 5, "Recovering Lost and Damaged Files".

Preserving Configuration and State Information

Even when you have safely stored backup copies of all configuration files and scripts needed for restoring the Oracle HSM software and file-system, it pays to preserve the current state of a failed system, if you can. Surviving configuration files and scripts may contain changes that were implemented since the complete configuration was last backed up. This can mean the difference between restoring the system to its almost its exact pre-failure state and merely getting close. Log and trace files contain information that helps restore files and clarifies the causes of failures. For this reason, you should preserve whatever remains, before you do anything else.

Save the Oracle HSM Configuration

1. If possible, log in to the file-system metadata server as root.

```
root@mds1:~#
```

2. Run the samexplorer command, create a SAMreport, and save the report in the directory that holds your backup configuration information. Use the command samexplorer *path*/*hostname*.*YYYYMMDD*.*hhmmz*.tar.gz, where *path* is the path to the chosen directory, *hostname* is the name of the Oracle HSM file system host, and *YYYYMMDD*.*hhmmz* is a date and time stamp.

   The default file name is /tmp/SAMreport.*hostname*.*YYYYMMDD*.*hhmmz*.tar.gz. In the example, we already have a directory for saving SAMreports, /zfs1/hsmcfg/. So we create the report in this directory:

```
root@mds1:~# samexplorer /zfs1/hsmcfg/server1.20160325.1659MST.tar.gz
     Report name:     /zfs1/hsmcfg/samhost1.20160325.1659MST.tar.gz
     Lines per file:  1000
     Output format:   tar.gz (default) Use -u for unarchived/uncompressed.

     Please wait.............................................
Please wait.............................................
Please wait......................................

The following files should now be ftp'ed to your support provider
as ftp type binary.

/zfs1/hsmcfg/samhost1.20160325.1659MST.tar.gz
```

3. Copy the `/etc/opt/SUNWsamfs/` directory and its contents to an independent file system.

   The `/etc/opt/SUNWsamfs/` directory may contain any or all of the following:

   - `mcf` (the master configuration file for the Oracle HSM file systems)
   - `archiver.cmd` (configures the archiving process)
   - `inquiry.conf` (lists the vendor and product identification strings that SCSI devices report in response to an inquiry command)
   - `scripts/*` (locally customized Oracle HSM scripts)
   - `defaults.conf` (overrides specified, default parameter values)
   - `diskvols.conf` (identifies disk storage that is used for archiving)
   - `hosts.`*`family-set-name`* (defines server and client host names and IP addresses for a shared file-system)
   - `hosts.`*`family-set-name`*`.local` (defines server and client host names and IP addresses for a shared file-system)
   - `preview.cmd` (customizes the priorities of archiving and staging requests for volumes that are not currently loaded)
   - `recycler.cmd` (customizes the recycling process)
   - `releaser.cmd` (customizes the releasing process)
   - `rft.cmd` (controls the Oracle HSM file transfer service)
   - `samfs.cmd` (defines file system mount parameters)
   - `stager.cmd` (customizes the staging process)
   - `samremote` (the SAM-Remote server configuration file)
   - *`family-set-name`* (a SAM-Remote client configuration file)
   - *`network-attached-library`* (a parameters file for a network-attached library

4. Back up all surviving library catalogs, including the historian catalog. For each catalog, use the command `dump_cat -V` *`catalog-file`*, where *`catalog-file`* is the path and name of the catalog file. Redirect the output to *`dump-file`* in an independent file system.

   We will use the output of the `dump_cat` file to rebuild the catalogs on a replacement system. In the example, we first dump the catalog data for `lib1` to the file `lib1cat1.dump` in a directory on the independent NFS-mounted file system `zfs1`. Then we dump the historian catalog:

```
root@mds1:~# dump_cat -V /var/opt/SUNWsamfs/catalog/lib1 >
/zfs1/hsmcfg/lib1cat1.dump
root@mds1:~# dump_cat -V /var/opt/SUNWsamfs/catalog/historian >
/zfs1/hsmcfg/historian1.dump
```

5. Copy system configuration files that were modified during Oracle HSM installation and configuration to an independent file system. These may include:

   ```
   /etc/
        syslog.conf
        system
        vfstab
   /kernel/drv/
        sgen.conf
        samst.conf
        samrd.conf
        sd.conf
        ssd.conf
        st.conf
   /usr/kernel/drv/dst.conf
   ```

6. Copy any custom shell scripts and `crontab` entries that you created as part of the Oracle HSM configuration to an independent file system.

   For example, if you created a `crontab` entry to manage creation of recovery points, you would save a copy now.

7. Create a `readme` file that records the revision level of the currently installed software. Include Oracle Oracle HSM, Solaris, and Solaris Cluster (if applicable). Save the file on an independent file system with the other recovery information.

8. If possible, save copies of downloaded Oracle Oracle HSM, Solaris, and Solaris Cluster packages on an independent file system.

   If you have the packages readily available, you can restore the software quickly, should it become necessary.

9. If you are recovering from the loss of a Oracle HSM server host, go to Chapter 3, "Restoring the Oracle HSM Configuration".

10. If you need to restore one or more Oracle HSM file systems, go to Chapter 4, "Recovering File Systems".

11. If you need to restore directories and files, go to Chapter 5, "Recovering Lost and Damaged Files".

3

Restoring the Oracle HSM Configuration

This chapter outlines the process of recovering the Oracle Hierarchical Storage Manager and StorageTek QFS Software and file-system configuration in the event that it is lost or corrupted, either in part or in its entirety. If the server host fails, the Oracle HSM software and file-system configurations may be lost, leaving file system data and metadata intact but inaccessible until the configuration information is restored. Success in this situation depends on your ability to salvage information from any files and directories that remain and on the thoroughness of your disaster preparations:

- If you have backup copies of your Oracle HSM configuration files and/or current SAMreports, use them to restore the configuration
- Otherwise, recreate the configuration using whatever information remains available.

Restoring the Configuration from Backup Copies and/or SAMreports

If you followed the procedures recommended in the *Oracle Hierarchical Storage Manager and StorageTek QFS Installation and Configuration Guide*, you can recover Oracle HSM software and file-system configurations using the procedure below.

Restore the Configuration from Backup Files or SAMreports

1. If you are restoring the configuration following a server host failure, resolve hardware issues, re-install operating systems, and re-install software as needed.
2. If a current backup copy of the root file system exists, restore the root file system and stop here.
3. Otherwise, log in to the file-system server host as `root`.

   ```
   root@solaris:~#
   ```

4. Mount any required file systems. Mount file systems that store backup Oracle HSM configuration files and any file systems that hold disk-archive copies of data files.

   In the example, we have been maintaining copies of the Oracle HSM server's Solaris configuration files in the subdirectory `sam_config` on the independent file system `zfs1`. So we create a mount point. We mount `zfs1`. We restore the `vfstab` file from the most recent copy in the `zfs1` file system. We create the needed mount points. Then we mount the file systems:

   ```
   root@solaris:~# cp /etc/vfstab /etc/vfstab.back
   root@solaris:~# mkdir /zfs1
   root@solaris:~# mount -F zfs /net/remote.example.com/zfs1/ /zfs1
   ```

```
root@solaris:~# cp /zfs1/sam_config/20140127/etc/vfstab /etc/vfstab
root@solaris:~# mkdir /diskvols
root@solaris:~# mkdir /diskvols/DISKVOL1
root@solaris:~# mkdir /diskvols/DISKVOL2
...
root@solaris:~# mount /diskvols/DISKVOL1
root@solaris:~# mount /diskvols/DISKVOL2
...
root@solaris:~#
```

5. If backup copies of the Oracle HSM configuration files are available, locate the most recent copies that are dated prior to the loss of the configuration.

   In the example, we have been maintaining copies of the Oracle HSM configuration files in the subdirectory `sam_config`, on the independent file system `/zfs1`. So the latest files are easy to find:

   ```
   root@solaris:~# ls /zfs1/sam_config/20140127/etc/opt/SUNWsamfs/
   archiver.cmd     defaults.conf    mcf              recycler.cmd     stager.cmd
   cfg_backups      diskvols.conf    mgmt_sched.conf  releaser.cmd     startup
   csn              inquiry.conf     notify.cmd       scripts          verifyd.cmd
   root@solaris:~# ls /zfs1/sam_config/20140127/etc/opt/SUNWsamfs/scripts
   archiver.sh    log_rotate.sh  nrecycler.sh   recycler.sh    save_core.sh
   sendtrap       ssi.sh
   root@solaris:~# ls /zfs1/sam_config/explorer/
   server1.20140430.1659MST.tar.gz   server1.20140114.0905MST.tar.gz
   server1.20110714.1000MST.tar.gz
   ```

6. If SAMreports were generated before the loss of the Oracle HSM configuration, locate the most recent.

7. If any QFS file systems are currently mounted, unmount them.

8. For each missing configuration file, copy an available backup file to the required location on the server that you are restoring.

   In the example, we restore all of the Oracle HSM configuration files and scripts from backup copies:

   ```
   root@solaris:~# cp /zfs1/sam_config/20140127/etc/opt/SUNWsamfs/*
   /etc/opt/SUNWsamfs/
   root@solaris:~# cp /zfs1/sam_config/20140127/etc/opt/SUNWsamfs/scripts/*
   /etc/opt/SUNWsamfs/scripts/
   root@solaris:~# cp /zfs1/sam_config/20140127/etc/opt/SUNWsamfs/startup/*
   /etc/opt/SUNWsamfs/startup/
   root@solaris:~# cp /zfs1/sam_config/20140127/etc/opt/SUNWsamfs/cfg_backups/*
   /etc/opt/SUNWsamfs/cfg_backups/
   root@solaris:~# cp /zfs1/sam_config/20140127/etc/opt/SUNWsamfs/csn/*
   /etc/opt/SUNWsamfs/csn/
   ```

9. If backup copies of the configuration files are not available, recreate them using the information contained in the most recent available SAMreport. Copy the content from the report, paste it into a text editor, and save it to the file and path indicated in the report.

   SAMreport files contain the full text of the Oracle HSM configuration files as they were at the time the report was created. They also list the directory where the file was located.

   In the example, we search the file `server1.20140127.SAMreport` for Oracle HSM master configuration file (`mcf`) information. We pipe the output of the `cat` command to the grep command and the regular-expression pattern `\/etc\/opt\/SUNWsamfs\/mcf`:

```
root@solaris:~# cat /zfs1/sam_config/explorer/server1.20140127.SAMreport | grep
\/etc\/opt\/SUNWsamfs\/mcf
...
------------------ /etc/opt/SUNWsamfs/mcf -------------------
server1# /bin/ls -l /etc/opt/SUNWsamfs/mcf
-rw-r--r--   1 root     root        1789 Feb  4 09:22 /etc/opt/SUNWsamfs/mcf

# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#-------------------  ---------  ---------  ---------  ------   -----------
hsmfs1               100        ms         hsmfs1    on
  /dev/dsk/c1t3d0s3    101        md         hsmfs1    on
  /dev/dsk/c1t4d0s5    102        md         hsmfs1    on
root@solaris:~#
```

We copy the output of the grep command, paste the output into the vi editor, and save the file to the correct name and location:

```
root@solaris:~# vi /etc/opt/SUNWsamfs/mcf
# Equipment           Equipment  Equipment  Family     Device   Additional
# Identifier          Ordinal    Type       Set        State    Parameters
#-------------------  ---------  ---------  ---------  ------   -----------
hsmfs1               100        ms         hsmfs1    on
 /dev/dsk/c1t3d0s3     101        md         hsmfs1    on
 /dev/dsk/c1t4d0s5     102        md         hsmfs1    on
:wq
root@solaris:~#
```

10. Restore the library catalogs from the dump-file data that you saved during the procedure "Save the Oracle HSM Configuration" on page 2-4. For each catalog, use the command `build_cat` *`catalog-dump-file catalog-file`*, where:

    - *`catalog-dump-file`* is the path and name of the file that you created with the `dump_cat` command.
    - *`catalog-file`* is the path and name of the restored catalog file.

    In the example, we rebuild the catalog for `lib1` using the data in the file `/zfs1/sam_config/20140513/catalogs/lib1cat.dump`:

```
root@solaris:~# build_cat /zfs1/sam_config/20140513/catalogs/lib1cat.dump \
/var/opt/SUNWsamfs/catalog/lib1cat
```

11. If you are recovering a system following hardware failure, go to Chapter 4, "Recovering File Systems".

12. If you are replacing one or more configuration files that were inadvertently deleted or incorrectly edited and if no hardware or file system changes have occurred, check the configuration files for errors by running the `sam-fsd` command.

    The `sam-fsd` is an initialization command that reads Oracle HSM configuration files. It will stop if it encounters an error:

```
root@solaris:~# sam-fsd
```

13. If the `sam-fsd` command finds an error in the `mcf` file, edit the file to correct the error and recheck as described in the preceding step.

    In the example below, `sam-fsd` reports an unspecified problem with a device:

```
root@solaris:~# sam-fsd
Problem in mcf file /etc/opt/SUNWsamfs/mcf for filesystem hsmfs1
sam-fsd: Problem with file system devices.
root@solaris:~#
```

14. If the `sam-fsd` command runs without error, the configuration files are correct. Proceed to the next step.

    The example is a partial listing of error-free output:

```
root@solaris:~# sam-fsd
Trace file controls:
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
              cust err fatal ipc misc proc date
              size    10M  age 0
sam-archiverd /var/opt/SUNWsamfs/trace/sam-archiverd
              cust err fatal ipc misc proc date module
              size    10M  age 0
sam-catserverd /var/opt/SUNWsamfs/trace/sam-catserverd
              cust err fatal ipc misc proc date module
              size    10M  age 0
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
root@solaris:~#
```

15. Tell the Oracle HSM software to read the `mcf` file and reconfigure itself accordingly. Use the command `samd config`.

```
root@solaris:~# samd config
Configuring SAM-FS
root@solaris:~#
```

16. If the `samd config` command reports errors in the `mcf` file, correct them. Then repeat the preceding step.
17. Remount the affected file systems.
18. Monitor file system operations.

Restoring the Configuration Without Backup Information

If you lack backup files or SAMreports, reconstruct the configuration using whatever information is available. Then proceed as for a new configuration. See the *Oracle Hierarchical Storage Manager and StorageTek QFS Installation and Configuration Guide* for instructions.

4

Recovering File Systems

This section outlines the recovery processes that you use when an entire Oracle HSM file system is corrupted or lost. The procedures vary, depending on the type of file system involved and the type of backup and recovery preparations that you have made. But there are two basic tasks that you have to perform:

- recreating the file system
- restoring directories and files.

Before you begin, please note: if you are recovering from the loss of an Oracle HSM metadata server, make sure that you have finished restoring the Oracle HSM configuration, as described in Chapter 3, before proceeding further. The procedures in this chapter assume that the Oracle HSM software is installed and configured as it was prior to the loss of the file system.

Recreating the File System

Before you can recover files and directories, you must have somewhere to put them. So the first step in the recovery process is to create an empty, replacement file system. Proceed as follows:

Recreate the File System Using Backup Configuration Files

1. Log in to the file-system metadata server as `root`.

   ```
   root@mds1:~#
   ```

2. Unmount the file system, if it is currently mounted. Use the command `umount` *`mount-point`*, where *`mount-point`* is the directory on which the file system is mounted.

   In the example, we unmount the file system `/hsm/hqfs1`:

   ```
   root@mds1:~# umount /hsm/hqfs1
   root@mds1:~#
   ```

3. Open the `/etc/opt/SUNWsamfs/mcf` file in a text editor. Check the hardware configuration. If you have had to change hardware, edit the file accordingly and save the changes.

   In the example, we replace the equipment identifiers for two failed disk devices with those of their replacements. Note that the equipment ordinals remain unchanged:

   ```
   root@mds1:~# vi /etc/opt/SUNWsamfs/mcf
   # Equipment              Equipment  Equipment  Family     Device  Additional
   ```

```
# Identifier            Ordinal    Type       Set        State   Parameters
#----------------------- ---------  ---------  ---------  ------  -------------
hqfs1                    100        ms         hqfs1      on
/dev/dsk/c1t3d0s3        101        md         hqfs1      on
/dev/dsk/c1t4d0s5        102        md         hqfs1      on
# Tape library
/dev/scsi/changer/c1t2d0 800        rb         lib800     on     .../lib800_cat
/dev/rmt/0cbn            801        li         lib800     on
/dev/rmt/1cbn            802        li         lib800     on
:wq
root@mds1:~#
```

4. Check the `mcf` file for errors. Use the command `sam-fsd`.

   The `sam-fsd` command is reads Oracle HSM configuration files and initializes the software. It will stop if it encounters an error:

   ```
   root@mds1:~# sam-fsd
   ```

5. If the `sam-fsd` command finds an error in the `mcf` file, edit the file to correct the error and recheck as described in the preceding step.

   In the example below, `sam-fsd` reports an unspecified problem with a device. This is probably a typo in an equipment identifier field:

   ```
   root@mds1:~# sam-fsd
   Problem in mcf file /etc/opt/SUNWsamfs/mcf for filesystem hqfs1
   sam-fsd: Problem with file system devices.
   ```

   Usually, such errors are the result of inadvertent typing mistakes. Here, when we open the `mcf` file in an editor, we find that we have typed a letter `o` instead of a 0 in the slice number part of the equipment name for device `102`, the second `md` device:

   ```
   root@mds1:~# vi /etc/opt/SUNWsamfs/mcf
   ...
   hqfs1                100        ms         hqfs1      on
   /dev/dsk/c0t0d0s0    101        md         hqfs1      on
   /dev/dsk/c0t3d0so    102        md         hqfs1      on
   ```

   So we correct the error, save the file, and recheck:

   ```
   root@mds1:~# vi /etc/opt/SUNWsamfs/mcf
   ...
   hqfs1                100        ms         hqfs1      on
   /dev/dsk/c0t0d0s0    101        md         hqfs1      on
   /dev/dsk/c0t3d0s0    102        md         hqfs1      on
   :wq
   root@mds1:~# sam-fsd
   ```

6. When the `sam-fsd` command runs without error, the `mcf` file is correct. Proceed to the next step.

   In the example, `sam-fsd` runs without error:

   ```
   root@mds1:~# sam-fsd
   Trace file controls:
   sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
   ...
   Would start sam-archiverd()
   Would start sam-stagealld()
   Would start sam-stagerd()
   Would start sam-amld()
   root@mds1:~#
   ```

7. Tell the Oracle HSM software to read the `mcf` file and reconfigure itself accordingly:

```
root@mds1:~# samd config
Configuring SAM-FS
root@mds1:~#
```

8. Create the replacement file system. Use the command `sammkfs` *`family-set-name`*, where *`family-set-name`* is the name of the file system.

   In the example, we recreate file system `hqfs1`:

```
root@mds1:~# sammkfs hqfs1
Building 'hqfs1' will destroy the contents of devices:
  /dev/dsk/c0t0d0s0
  /dev/dsk/c0t3d0s0
Do you wish to continue? [y/N]yes
total data kilobytes       = ...
root@mds1:~#
```

9. Recreate the mount point directory for the file system, if necessary.

   In the example, we recreate the directory `/hsm/hqfs1`:

```
root@mds1:~# mkdir /hsm
root@mds1:~# mkdir /hsm/hqfs1
root@mds1:~#
```

10. Back up the operating system's `/etc/vfstab` file.

```
root@mds1:~# cp /etc/vfstab /etc/vfstab.backup
root@mds1:~#
```

11. Open the `/etc/vfstab` file in a text editor. If the `/etc/vfstab` file does not contain mount parameters for the file system that you are restoring, you will have to restore the mount parameters.

    In the example, the Oracle HSM server is installed on a replacement host. So the file contains no mount parameters for the file system that we are restoring, `hqfs1`:

```
root@mds1:~# vi /etc/vfstab
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  ---------------------
/devices   -        /devices  devfs   -     no       -
/proc      -        /proc     proc    -     no       -
...
```

12. If possible, when you must restore mount parameters, open a backup copy of the original `/etc/vfstab` file and copy the required line into the current `/etc/vfstab` file. When the changes are complete, save the file and close the editor.

    In the example, we have a backup copy, `/zfs1/sam_config/20161027/etc/vfstab`. So we copy the line for the `hqfs1` file system from the backup copy and paste it into the current `/etc/vfstab` file:

```
root@mds1:~# vi /zfs1/sam_config/20161027/etc/vfstab.20161027
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  ---------------------
```

```
/devices   -        /devices  devfs   -     no       -
/proc      -        /proc     proc    -     no       -
...
hqfs1      -        /hqfs1    samfs   -     yes      stripe=1,bg
:q
root@mds1:~# vi /etc/vfstab
#File
#Device    Device   Mount     System  fsck  Mount    Mount
#to Mount  to fsck  Point     Type    Pass  at Boot  Options
#--------  -------  --------  ------  ----  -------  ---------------------
/devices   -        /devices  devfs   -     no       -
/proc      -        /proc     proc    -     no       -
...
hqfs1      -        /hqfs1    samfs   -     yes      stripe=1,bg
:wq
root@mds1:~#
```

13. Mount the file system.

    In the example, we mount the file system hqfs1:

    ```
    root@mds1:~# mount /hsm/hqfs1
    root@mds1:~#
    ```

14. Now, start restoring directories and files.

Restoring Directories and Files

Once you have recreated the base file system, you can start to restore directories and files. There are two possible approaches:

- Restoring files and directories from a samfsdump (qfsdump) recovery point file is by far the best option, if you have created and safely stored recovery points on a regular basis.

  This approach returns the file system to full functionality immediately, because it restores the file-system metadata. An archiving file system can immediately access data on archival media and stage files back to the disk cache, either immediately or as-needed, when users access files. Files are restored with their original attributes.

  If the recovery point contains data as well as metadata, this approach is also the only way to restore stand-alone (non-archiving) file systems that are not backed up by third-party applications.

- Restoring files and directories from archival media without a recovery point file using a recovery script and the Oracle HSM star utility.

Restoring Files and Directories from a `samfsdump` (`qfsdump`) Recovery Point File

Whenever possible, you should base file-system recovery efforts on the most recent available recovery point file. This approach is by far the fastest, most reliable, most thorough, and least labor-intensive way of recovering from the failure of a Oracle HSM file system. So, if a recovery point file exists, proceed as follows:

Restore the Lost File System from a Recovery Point File

1. Log in to the file-system metadata server as root.

   ```
   root@mds1:~#
   ```

2. If you have not already done so, stop archiving and recycling using the procedures in "Stopping Archiving and Recycling Processes" on page 2-1.

3. Identify the most recent available recovery point file.

   In the example, we have been creating dated recovery point files for the file system `hqfs1` in a well-known location, the subdirectory `hqfs1_recovery` on the independent file system `/zfs1`. So the latest file, `20161024`, is easy to find:

   ```
   root@mds1:~# ls /zfs1/hqfs1_recovery/
   20161021    20161022    20161023    20161024
   root@mds1:~#
   ```

4. Change to the mount-point directory for the recreated file system.

   In the example, the recreated file system is mounted at `/hsm/hqfs1`:

   ```
   root@mds1:~# cd /hsm/hqfs1
   root@mds1:~#
   ```

5. Restore the entire file system relative to the current directory. Use the command `samfsrestore -T -f` *`recovery-point-file`* `-g` *`logfile`* or the QFS-only command `qfsrestore -T -f` *`recovery-point-file`* `-g` *`logfile`*, where:

   - `-T` displays recovery statistics when the command terminates, including the number of files and directories processed and the number of errors and warnings.
   - `-f` *`recovery-point-file`* specifies the path and file name of the selected recovery point file.
   - `-g` *`logfile`* creates a list of the directories and files that were online when the recovery point was created and saves the list to the file specified by *`logfile`*.

     If you are restoring an archiving file system, this file can be used to automatically stage files from archival media, so that the disk cache is in the same state as it was at the time that the recovery point was created.

   In the example, we restore the file system `hqfs1` from the recovery point file `/zfs1/hqfs1_recovery/20161024`. We log the online files in the file `/root/20161024.log`:

   ```
   root@mds1:~# samfsrestore -T -f /zfs1/hqfs1_recovery/20161024 -g
   /root/20161024.log
         samfsdump statistics:
                   Files:              52020
                   Directories:        36031
                   Symbolic links:     0
                   Resource files:     8
                   File segments:      0
                   File archives:      0
                   Damaged files:      0
                   Files with data:    24102
                   File warnings:      0
                   Errors:             0
                   Unprocessed dirs:   0
                   File data bytes:    0
   root@mds1:~#
   ```

6. If you have restored a standalone (non-archiving) file system, the file-system metadata and file data that were saved in the recovery-point file have been restored. Stop here.

7. Otherwise, restage archived files if required.

Restage Archived Files If Required

1. In most cases, do not restage files from archival media to disk following a file system recovery. Let users stage files as needed, by accessing them.

   This approach automatically prioritizes staging according to user needs. It maximizes the availability of the file system at a time when it may have been offline for some time.

   Only immediately required files are staged. So the total staging effort is spread over a period of time. This helps to insure that file system resources, such as drives, are always available for high priority tasks, such as archiving new files and staging urgently required user data.

   This approach also reduces the administrative effort associated with recovery.

2. If you must restage the files that were resident in the disk cache prior to a failure, use the command `/opt/SUNWsamfs/examples/restore.sh` *`logfile`*, where *`logfile`* is the path and file name of the log file that you created with the `-g` option of the `samfsrestore` (`qfsrestore`) command.

   The `restore.sh` script stages the files listed in the log file. These are the files that were online when the `samfsrestore` (`qfsrestore`) recovery point file was created.

   If thousands of files need to be staged, consider splitting the log file into smaller files. Then run the `restore.sh` script with each file in turn. This spreads the staging effort over a period of time and reduces interference with archiving and user-initiated staging.

3. Now identify damaged files and locate replacement copies.

Identify Damaged Files and Locate Replacement Copies

The `samfsrestore` process restores a copy of the file-system metadata from a recovery point file, so that you can find the corresponding data on tape and restore it to its proper locations in the file system. Recovery point files are created prior to the loss of the file system, however. So, inevitably, some of the metadata typically points to data locations that have changed since the recovery point was created. The file system has a record of these files but cannot locate their contents. So it sets the *damaged* flag on each such file.

In some cases, the data for a damaged file may, indeed, be lost. But in other cases, the restored metadata is simply out of date. The restored file system may not be able to find data for files that were archived or migrated after the recovery point was created simply because the restored metadata does not record a current location. In these cases, you may be able to undamage the files by locating the data yourself and then updating the restored metadata.

To locate missing data, update metadata, and undamage files, use the archiver log and media-migration log files (if any). Proceed as follows:

1. If you have not already done so, log in to the file-system metadata server as `root`.

   ```
   root@mds1:~#
   ```

2. Identify the most recent available archiver log file.

   If the archiver log on the server is still available, it is likely to contain the most recent information. Otherwise, you will need to use a backup copy.

In the example, the archiver log file `hqfs1.archiver.log` is on the server in the `/var/adm/` subdirectory. We also have dated archiver log file copies in a well-known location, the subdirectory `hqfs1_recovery/archlogs` on the independent file system `/zfs1`. So the we have both the latest file, `hqfs1.archiver.log`, and a recent backup, `20161024`:

```
root@mds1:~# dir /var/adm/*.archiver.log
hqfs1.archiver.log
root@mds1:~# dir /zfs1/hqfs1_recovery/archivelogs
20161022    20161023    20161024
root@mds1:~#
```

3. If files were recently migrated to replacement media, locate the migration logs as well.

   Media migration logs are created for each source volume in the logging directory specified by the `migrationd.cmd` file. Logs are named *`media-type.vsn`*, where *`media-type`* is one of the two-digit codes described in Appendix B, "Glossary of Equipment Types" and *`vsn`* is the six-character, alphanumeric Volume Serial Number of the source volume.

   The format of media-migration logs contain the same recovery information as archiver logs and can be used in the same fashion. For a description of the few format differences, see Appendix A, "Understanding Archiver and Migration Logs".

4. In the newly restored file system, identify any damaged files. Use the command `sfind` *`mountpoint`* `-damaged`, where *`mountpoint`* is the directory where the recovered file system is mounted.

   In the example, we start the search in the directory `/hsm/hqfs1` and find six damaged files:

```
root@mds1:~# sfind /hsm/hqfs1 -damaged
./genfiles/ay0
./genfiles/ay1
./genfiles/ay2
./genfiles/ay5
./genfiles/ay6
./genfiles/ay9
root@mds1:~#
```

5. Search the most recent copy of the archiver log for entries relating to each of the damaged files. Use the command `grep "`*`file-name-expression`*`"` *`archiver-log`*, where *`file-name-expression`* is a regular expression that matches the damaged file and *`archiver-log`* is the path and name of the archiver log copy that you are examining.

   In the example, we use the regular expression `genfiles\/ay0` to search the most recent log file for entries relating to the file `genfiles/ay0`:

```
root@mds1:~# grep "genfiles\/ay0 " /var/adm/hqfs1.archiver.log
```

6. When you find an entry for a file, note the media type, volume serial number, and position of the archive (`tar`) file where the data file is archived. Also note the file type, since this will affect how you restore the file.

   In the example, we locate an entry for the file `genfiles/ay0`. The log entry shows that it was archived (`A`) on October 24, 2016 at 9:49 PM using LTO (`li`) volume `VOL012`. The file is stored in the tape archive file located at hexadecimal position 0x78 (`78`). The file is a regular file, type `f`:

```
root@mds1:~# grep "genfiles\/ay0 " /var/adm/hqfs1.archiver.log
A 2016/10/24 21:49:15 li VOL012 SLOT12 allsets.1 78.1 hqfs1 7131.14 8087
genfiles/ay0 f 0 51
root@mds1:~#
```

For a full explanation of the fields in archiver log entries, see Appendix A, "Understanding Archiver and Migration Logs".

7. If you do not find an entry for a damaged file in the current archiver log copy, repeat the search using any backup archive logs that were created after the recovery point file was created.

   Archiver logs are rolled over frequently. So, if you retain multiple archiver log copies, you may be able to recover damaged files using archive copies that were made before the period covered by the current archiver log.

8. Next, look for files that were archived after the recovery point was created.

Look for Missing Files that Were Archived After the Recovery Point Was Created

The samfsrestore process restores a copy of the file-system metadata from a recovery point file, so that you can find the corresponding file-system data on tape and restore it to its proper locations in the file system. Recovery point files are created prior to the loss of the file system, however. They cannot contain metadata for files created and archived thereafter.

Typically, some files are archived after the last recovery point was created and prior to the loss of a file system. Since the metadata for these files are not in the recovery point file, samfsrestore cannot recover them, even as damaged files. File data does, however, reside on archival media. So you can recreate the metadata and recover the files to their proper place in the file system using the archive logs. If files were migrated to replacement media prior to the loss of the file system, you can use media-migration logs as well.

1. If you have not already done so, log in to the file-system metadata server as root.

   ```
   root@mds1:~#
   ```

2. Identify the most recent available archiver log file.

   If the archiver log on the server is still available, it is likely to contain the most recent information. Otherwise, you will need to use a backup copy.

   In the example, the archiver log file hqfs1.archiver.log is on the server in the /var/adm/ subdirectory. We also have dated archiver log file copies in a well-known location, the subdirectory hqfs1_recovery/archlogs on the independent file system /zfs1. So the we have both the latest file, hqfs1.archiver.log, and a recent backup, 20161024:

   ```
   root@mds1:~# dir /var/adm/*.archiver.log
   hqfs1.archiver.log
   root@mds1:~# dir /zfs1/hqfs1_recovery/archivelogs
   20161022    20161023    20161024
   root@mds1:~#
   ```

3. If files were recently migrated to replacement media, locate the migration logs as well.

   Media migration logs are created for each source volume in the logging directory specified by the migrationd.cmd file. Logs are named *media-type*.*vsn*, where *media-type* is one of the two-digit codes described in Appendix B, "Glossary of Equipment Types" and *vsn* is the six-character, alphanumeric Volume Serial

Number of the source volume.

The format of media-migration logs contain the same recovery information as archiver logs and can be used in the same fashion. For a description of the few format differences, see Appendix A, "Understanding Archiver and Migration Logs".

4. Search the most recent copy of the archiver log for entries that were made after the recovery point was created. Use the command `grep "`*`time-date-expression`*`"` *`archiver-log`*, where *`time-date-expression`* is a regular expression that matches the date and time where you want to start searching and *`archiver-log`* is the path and name of the archiver log copy that you are examining.

   In the example, we lost the file system at 2:02 AM on October 25, 2016. The last recovery point file was made at 2:10 AM on October 24, 2016. So we use the regular expression `^A 2016\/10\/2[45]` to search the most recent log file for archived files that were logged on October 24 or 25:

```
root@mds1:~# grep "^A 2016\/10\/2[45]" /var/adm/hqfs1.archiver.log
```

5. When you find an entry for an archived copy of an unrestored file, note the path, name, file type, media type, and location information.

   File types are listed as `f` for regular files, `R` for removable-media files, or `S` for a data segment in a segmented file. The media type is a two-character code (see Appendix B, "Glossary of Equipment Types").

   To locate the backup copy, you need the volume serial number of the media volume that stores the copy. If the copy is stored on sequential-access media, such as magnetic tape, also note the hexadecimal value that represents the starting position of the archive (`tar`) file. If the copy is stored on random-access media, such as archival disk, note the path and file name of the `tar` file relative to the volume serial number. Finally, if the file is segmented, note the segment length.

   In the example below, the archiver log entries show that the following files were archived after the last recovery point was created:

```
root@mds1:~# grep "^A 2016\/10\/2[45]" /var/adm/hqfs1.archiver.log
A 2016/10/24 10:43:18 li VOL002 all.1 111.1 hqfs1 1053.3 69 genfiles/hops f 0 0
A 2016/10/24 10:43:18 li VOL002 all.1 111.3 hqfs1 1051.1 104 genfiles/anic f 0
0
A 2016/10/24 13:09:05 li VOL004 all.1 212.1 hqfs1 1535.2 1971 genfiles/genA0 f
0 0
A 2016/10/24 13:09:06 li VOL004 all.1 212.20 hqfs1 1534.2 1497 genfiles/genA9 f
0 0
A 2016/10/24 13:10:15 li VOL004 all.1 212.3f hqfs1 1533.2 6491 genfiles/genA2 f
0 0
A 2016/10/24 13:12:25 li VOL003 all.1 2.5e hqfs1 1532.2 17717 genfiles/genA13 f
0 0
A 2016/10/24 13:12:28 li VOL003 all.1 2.7d hqfs1 1531.2 14472 genfiles/genA4 f
0 0
A 2016/10/24 13:12:40 li VOL003 all.1 2.9c hqfs1 1530.2 19971 genfiles/genA45 f
0 0
A 2016/10/24 21:49:15 dk DISKVOL1/f2 all.1 2.2e9 hqfs1 1511.2 8971
socfiles/spcC4 f 0 0
A 2016/10/24 21:49:15 dk DISKVOL1/f2 all.1 2.308 hqfs1 1510.2 7797
spcfiles/spcC5 f 0 0
A 2016/10/24 14:01:47 li VOL013 all.1 76a.1 hqfs1 14.5 10485760 bf/dat011/1 S 0
51
A 2016/10/24 14:04:11 li VOL013 all.1 76a.5002 hqfs1 15.5 10485760 bf/dat011/2
S 0 51
```

```
A 2016/10/24 14:06:24 li VOL013 all.1 1409aa4.1 hqfs1 16.5 184 bf/dat011/3 S 0
51
A 2016/10/24 18:28:51 li VOL036 all.1 12d.1 hqfs1 11731.1 89128448  rf/rf81 f 0
210
A 2016/10/24 18:28:51 li VOL034 all.1 15f.0 hqfs1 11731.1 525271552 rf/rf81 f 1
220
root@mds1:~#
```

We note the following information:

- Eight regular (type `f`) files are archived (`A`) on LTO (`li`) media: `genfiles/hops` and `genfiles/anic` at position `0x111` on volume `VOL002`, `genfiles/genA0`, `genfiles/genA9` and `genfiles/genA2` at position `0x212` on volume `VOL004`, and `genfiles/genA13`, `genfiles/genA4`, and `genfiles/genA45` at position `0x212` on volume `VOL003`.
- Two regular (type `f`) files are archived (`A`) on disk (`dk`) media: `spcfiles/spcC4` and `spcfiles/spcC5` in archive file `DISKVOL1 \f2` on volume `DISKVOL1`.
- One, three-part, segmented (type `S`) file is archived on LTO (`li`) media: `bf/dat011`, in two segments starting at position `0x76a` and one segment starting at position `1409aa4` on volume `VOL013`. Segment `/1` is `10485760` bytes long, segment `/2` is `10485622` bytes, and segment `/3` is `184` bytes.
- One, regular (type `f`), volume overflow file archived (`A`) on LTO (`li`) media: `rf/rf81`, starting at position `0x12d` on volume `VOL036` and continuing at position `0x15f` on volume `VOL034`.

For a full explanation of the fields in archiver log entries, see Appendix A, "Understanding Archiver and Migration Logs".

6. Repeat the search using any backup archive logs that were created after the recovery point file was created.

   Archiver logs are rolled over frequently. So, if you retain multiple archiver log copies, you may be able to recover damaged files using archive copies that were made before the period covered by the current archiver log.

7. Now restore the damaged and/or missing files.

Restore the Damaged and/or Missing Files

Given the media volume and the position of an archive (`tar`) file on the media, restoring a missing or damaged file is simply a matter of accessing the `tar` file and extracting the required data file. When the archive files reside on archival disk devices, this is simple, because the `tar` files reside in randomly accessible directories under a file-system mount point. When the `tar` file resides on high-capacity, sequential-access media like tape, however, there is an added complication: we cannot normally extract the required data file from the archive file until the latter is staged to a random-access disk device. Since archive files can be large, this can be time-consuming and awkward in a recovery situation. So the procedures below take advantage of the Oracle HSM command `request`, which reads the archive files into memory and makes them available as if they were being read from disk.

Restore as many damaged and missing regular files as you can. For each file, proceed as follows:

1. Start by recovering regular files that do not span volumes. Use the procedure"Restore Lost and Damaged Regular Files" on page 5-3.
2. Next, recover the segmented files. Use the procedure "Restore Lost and Damaged Segmented Files" on page 5-6.

3. Then restore the regular files that do span volumes. Use the procedure "Restore Lost and Damaged Volume Overflow Files" on page 5-9.

4. Once you have restored all missing and damaged files that have copies, re-enable archiving by removing `wait` directives from the `archiver.cmd` file. Re-enable recycling by removing `-ignore` parameters from the `recycler.cmd` file.

   The file system is as close to its original condition as possible. Files that are still damaged or missing cannot be recovered.

5. Once you have restored all missing and damaged files that have copies, go to "Restoring Archiving File Systems to Normal Operation" on page 6-1.

Restoring Files and Directories from Archival Media without a Recovery Point File

If you must recover a file system directly from the archival media, without the assistance of a recovery point file, you can do so. Proceed as follows:

1. If you are trying to restore files from optical media, stop here and contact Oracle support services for assistance.

2. Disable Network File System (NFS) sharing for the file system.

3. Disable archiving and recycling. Use the method outlined in "Stopping Archiving and Recycling Processes" on page 2-1.

4. Reserve a tape drive for recovery. Use the command `samcmd unavail` *`drive-equipment-number`*, where *`drive-equipment-number`* is the equipment ordinal number assigned to the drive in the `/etc/opt/SUNWsamfs/mcf` file.

   The `samcmd unavail` command makes the drive unavailable to archiving, staging and releasing processes. In the example, we reserve drive `804`

   ```
   root@mds1:~# samcmd unavail 804
   root@mds1:~#
   ```

5. Copy the file `/opt/SUNWsamfs/examples/tarback.sh` to an alternate location, such as `/tmp`.

   The `tarback.sh` file is an executable script that restores files from a specified set of media volumes. The script runs the command `star -n` against each archive (`tar`) file on each volume. When a backup copy on tape has no corresponding file in the file system or when the copy on tape is newer than the corresponding file in the file system, `star -n` restores the copy.

   In the example, we copy the script to `/tmp`:

   ```
   root@mds1:~# cp /opt/SUNWsamfs/examples/tarback.sh /tmp/tarback.sh
   root@mds1:~#
   ```

6. Open the copy of the `tarback.sh` file in a text editor.

   In the example, we use the `vi` editor:

   ```
   root@mds1:~# vi /tmp/tarback.sh
   #!/bin/sh
   #   script to reload files from SAMFS archive tapes
   STAR="/opt/SUNWsamfs/sbin/star"
   LOAD="/opt/SUNWsamfs/sbin/load"
   UNLOAD="/opt/SUNWsamfs/sbin/unload"
   EQ=28
   TAPEDRIVE="/dev/rmt/3cbn"
   # BLOCKSIZE is in units of 512 bytes (e.g. 256 for 128K)
   BLOCKSIZE=256
   ```

```
MEDIATYPE="lt"
VSN_LIST="VSNA VSNB VSNC VSNZ"
...
```

7. If the Oracle HSM utilities `star`, `load`, and `unload` are installed in non-standard locations, edit the default command paths in the copy of the `tarback.sh` file.

   In the example, all utilities are installed in the default locations, so no edits are needed:

   ```
   root@mds1:~# vi /tmp/tarback.sh
   #!/bin/sh
   #   script to reload files from SAMFS archive tapes
   STAR="/opt/SUNWsamfs/sbin/star"
   LOAD="/opt/SUNWsamfs/sbin/load"
   UNLOAD="/opt/SUNWsamfs/sbin/unload"
   ...
   ```

8. In the copy of the `tarback.sh` file, locate the variable `EQ`. Set its value to the equipment ordinal number of the drive that you reserved for recovery use.

   In the example, we set `EQ=804`:

   ```
   root@mds1:~# vi /tmp/tarback.sh
   #!/bin/sh
   #   script to reload files from SAMFS archive tapes
   STAR="/opt/SUNWsamfs/sbin/star"
   LOAD="/opt/SUNWsamfs/sbin/load"
   UNLOAD="/opt/SUNWsamfs/sbin/unload"
   EQ=804
   ...
   ```

9. In the copy of the `tarback.sh` file, locate the variable `TAPEDRIVE`. Set its value to the raw path to the device, enclosed in double quotation marks.

   In the example, the raw path to device `804` is `/dev/rmt/3cbn`:

   ```
   root@mds1:~# vi /tmp/tarback.sh
   #!/bin/sh
   #   script to reload files from SAMFS archive tapes
   STAR="/opt/SUNWsamfs/sbin/star"
   LOAD="/opt/SUNWsamfs/sbin/load"
   UNLOAD="/opt/SUNWsamfs/sbin/unload"
   EQ=804
   TAPEDRIVE="/dev/rmt/3cbn"
   ...
   ```

10. In the copy of the `tarback.sh` file, locate the variable `BLOCKSIZE`. Set its value to the number of 512-byte units in the desired block size.

    In the example, we want a 256-kilobyte block size for the LTO drive. So we specify `512`:

    ```
    LOAD="/opt/SUNWsamfs/sbin/load"
    UNLOAD="/opt/SUNWsamfs/sbin/unload"
    EQ=804
    TAPEDRIVE="/dev/rmt/3cbn"
    BLOCKSIZE=512
    ...
    ```

11. In the copy of the tarback.sh file, locate the variable MEDIATYPE. Set its value to the two-character media-type code that Appendix B lists for the type of media that the drive supports. Enclose the media type in double quotation marks.

    In the example, we are using an LTO-4 drive. So we specify li:

    ```
    EQ=804
    TAPEDRIVE="/dev/rmt/3cbn"
    BLOCKSIZE=512
    MEDIATYPE="li"
    ...
    ```

12. In the copy of the tarback.sh file, locate the variable VSN_LIST. As its value, supply a space-delimited list of the volume serial numbers (VSNs) that identify tapes that might contain backup copies of your files. Enclose the list in double quotation marks.

    In the example, we specify volumes VOL002, VOL003, VOL004, VOL013, VOL034, and VOL036:

    ```
    EQ=804
    TAPEDRIVE="/dev/rmt/3cbn"
    BLOCKSIZE=512
    MEDIATYPE="lt"
    VSN_LIST="VOL002 VOL003 VOL004 VOL013 VOL034 VOL036"
    ...
    ```

13. Save the copy of the tarback.sh file. Close the editor.

    ```
    EQ=804
    TAPEDRIVE="/dev/rmt/3cbn"
    BLOCKSIZE=512
    MEDIATYPE="lt"
    VSN_LIST="VOL002 VOL003 VOL004 VOL013 VOL034 VOL036"
    ...
    :wq
    root@mds1:~#
    ```

14. Execute the /tmp/tarback.sh script.

    ```
    root@mds1:~# /tmp/tarback.sh
    ```

15. For each restored file, recreate user and group ownership, modes, extended attributes, and access control lists (ACLs), as necessary.

    The /tmp/tarback.sh script cannot restore these types of metadata.

16. Once you have run the /tmp/tarback.sh script and finished recovering files, go to "Restoring Archiving File Systems to Normal Operation" on page 6-1.

5

Recovering Lost and Damaged Files

This chapter outlines procedures for restoring individual files to the file system. It covers the following tasks:

- recovering files using a recovery point file
- recovering files using log entries
- recovering damaged archive copies.

Recovering Files Using a Recovery Point File

A recovery point file is the fastest, most reliable, most thorough, and least labor-intensive way of recovering lost or damaged files. So, if a recovery point file is available, proceed as follows:

1. Log in to the file-system metadata server as `root`.

   ```
   root@mds1:~#
   ```

2. If you have not already done so, stop archiving and recycling using the procedures in "Stopping Archiving and Recycling Processes" on page 2-1

3. In the target file system, create a temporary recovery directory to hold the recovered files.

   In the example, we create the temporary directory `restore` under the mount point for the recreated file system, `/hsm/hqfs1`:

   ```
   root@mds1:~# mkdir /hsm/hqfs1/restore
   ```

4. Keep the archiver from archiving from the temporary directory. Use the command `archive -r -n` *`directory`*, where:

   - `-r -n` recursively disable archiving of files that reside in or under the specified directory.
   - *`directory`* is the path and directory name of the temporary recovery directory.

   ```
   root@mds1:~# archive -r -n /hsm/hqfs1/restore
   ```

5. Change to the temporary recovery directory.

   ```
   root@mds1:~# cd /hsm/hqfs1/restore
   ```

6. Identify the most recent available recovery point file.

In the example, we have been creating dated recovery point files for the file system hqfs1 in a well-known location, the subdirectory hqfs1_recovery on the independent file system /zfs1. So the latest file, 20161024, is easy to find:

```
root@mds1:~# dir /zfs1/hsm/hqfs1_recovery/
20161021    20161022    20161023    20161024
root@mds1:~#
```

7. Make sure that the file that you need to recover is in the recovery point file. Search for the required file in the output of the command samfsrestore -t -f *recovery-point*, where:

   - -t displays a table of contents.
   - -f *recovery-point-file* specifies the path and file name of the selected recovery point file.

   In the example, we are trying to recover the file genw445. So we run the command samfsrestore -t with the file /zfs1/hsm/hqfs1_recovery/20161024. To simplify the search, we pipe the output of samfsrestore -t to the Solaris grep command and the regular expression genw445:

```
root@mds1:~# samfsrestore -t -f /zfs1/hsm/hqfs1_recovery/20161024 | grep
"genw445"
./genfiles/genw445
root@mds1:~#
```

8. Restore the file's inode information to the current directory. Use the command samfsrestore -f *recovery-point file*, where:

   - -f *recovery-point-file* specifies the path and file name of the selected recovery point file.
   - *file* specifies the exact path and name that the recovery point file lists for the file that you want to recover.

   In the example, we recover ./genfiles/genw445 from the recovery point file /zfs1/hsm/hqfs1_recovery/20161024:

```
root@mds1:~# samfsrestore -f /zfs1/hsm/hqfs1_recovery/20161024
./genfiles/genw445
root@mds1:~#
```

9. Make sure that the file has been restored correctly. Use the command sls -D *file*, where *file* specifies the path and name of the file relative to the temporary recovery directory.

   In the example, the file genfiles/genw445 has been recovered to the temporary directory /hsm/hqfs1/restore/:

```
root@mds1:~# sls -D genfiles/genw445
genfiles/genw445:
  mode: -rw-r--r--    links:   1  owner: data        group: hqfs1
  length:     14975  inode:    25739.1
offline; archdone;
copy 1: ---- Mar  4 11:55 8ae.1 xt 000000
copy 2: ---- Mar  4 15:51 cd3.7f57 xt 000000
  access:      Mar  4 11:55  modification: Mar  4 21:50
  changed:     Mar  4 11:50  attributes:   Mar  4 21:50
  creation:    Mar  4 11:50  residence:    Mar  4 21:50
root@mds1:~#
```

10. If the file has been correctly restored, move it to the correct location in the file system.

    In the example, we move the file `genw445` from the temporary, working directory `/hsm/hqfs1/restore/genfiles/` to its original location in `/hsm/hqfs1/genfiles/`:

    ```
    root@mds1:~# mv -f genfiles/genw445 /hsm/hqfs1/genfiles/genw445
    root@mds1:~#
    ```

11. Repeat this procedure until all missing files have been recovered.

12. Finish the recovery procedure. Go to "Restoring Archiving File Systems to Normal Operation" on page 6-1.

Recovering Files Using Log Entries

Recovering files with an archiver log and/or media-migration logs is always a tedious and labor-intensive process, if more than a few files are involved. So, whenever possible, use the procedures in this section only when a recovery point cannot restore the file that you need.

While the overall process for recovering files from archival media is essentially the same in all cases, details can differ for different types of file. So select the procedure intended for the type of file that you are restoring:

- restoring lost and damaged regular files
- restoring lost and damaged segmented files
- restore lost and damaged volume overflow files.

Note that files may not be restored to the precise location that you expect when you recover a copy from the media. Files are restored to their location at the time when the archive copy was made. So files that were subsequently moved are not restored to the directory where they were when they were lost.

Restore Lost and Damaged Regular Files

For each file that you need to recover, proceed as follows:

1. If you have not already done so, log in to the file-system metadata server as `root`.

   ```
   root@mds1:~#
   ```

2. If you have not already done so, stop archiving and recycling using the procedures in "Stopping Archiving and Recycling Processes" on page 2-1

3. Change to the root directory of the file system that you are restoring.

   Oracle HSM archive files store copies relative to the file-system root directory. So to restore them to their original locations, we want to restore them from the root directory.

   In the example, we change to the root of the `hqfs1` file system:

   ```
   root@mds1:~# cd /hsm/hqfs1
   root@mds1:~#
   ```

4. If you have an archiver log for the period when the regular file was last archived, find the most recent entry for the file.

   In the first example, we look for an entry for the regular (type **`f`**) file `genA0`:

   ```
   A 2016/10/24 13:09:05 li VOL004 all.1 212.1 hqfs1 1535.2 1971 genfiles/genA0 f
   ```

```
0 0
```

In the second example, we look for an entry for the regular (type `f`) file `spcC4`:

```
A 2016/10/24 21:49:15 dk DISKVOL1/f2 all.1 2.2e9 hqfs1 1511.2 8971
socfiles/spcC4 f 0 0
```

5. Once you have located a log entry for a required file, note the media type, the volume serial number of the media, and the path and name of the file relative to the root directory of the file system.

   In the first example, the file `genA0` resides on an LTO (`li`) tape volume with the Volume Serial Number (VSN) `VOL004`. The file was originally stored in the file system directory `/hsm/hqfs1/genfiles/`:

   ```
   A 2016/10/24 13:09:05 li VOL004 all.1 212.1 hqfs1 1535.2 1971 genfiles/genA0 f
   0 0
   ```

   In the second example, the file `spcC4` resides in a disk archive (`dk`) with the volume serial number `DISKVOL1`. The file was originally stored in the file system directory `/hsm/hqfs1/socfiles/`:

   ```
   A 2016/10/24 21:49:15 dk DISKVOL1/f2 all.1 2.2e9 hqfs1 1511.2 8971
   socfiles/spcC4 f 0 0
   ```

6. If a required file resides on sequential-access media, such as magnetic tape, also note the hexadecimal value that represents the starting position of the archive (`tar`) file.

   In the example, file `genA0` resides on tape starting at position 0x212 (`212`):

   ```
   A 2016/10/24 13:09:05 li VOL004 all.1 212.1 hqfs1 1535.2 1971 genfiles/genA0 f
   0 0
   ```

7. If a required file resides on random-access media, such as archival disk, also note the path and file name of the `tar` file relative to the volume serial number.

   In the example, file `spcC4` resides in the `f2` subdirectory immediately under the root directory of disk volume `DISKVOL1`:

   ```
   A 2016/10/24 21:49:15 dk DISKVOL1/f2 all.1 2.2e9 hqfs1 1511.2 8971
   socfiles/spcC4 f 0 0
   ```

8. If the file that you are restoring is archived on disk media, extract the archive copy of the missing or damaged file from the `tar` file on the disk volume. Use the command `star -xv -f` *`tarfile file`*, where:

   - *`tarfile`* is the name of the archive file
   - *`file`* is the path—relative to the file-system root directory—and name of the file that you need to restore.

   The star command is an enhanced Oracle HSM version of GNU `tar` that restores specified files from the archive file.

   In the example, we extract the data file `socfiles/spcC4` from the `tar` file `DISKVOL1/f2`. The file is restored to `/hqfs1/socfiles/spcC4`:

   ```
   root@mds1:~# star -xvf DISKVOL1/f2 socfiles/spcC4
   ```

9. If you have restored the required file from a disk archive, continue to restore lost and damaged regular files until all required files have been restored.

10. If the file that you are restoring is archived on removable media, such as magnetic tape, create a directory in the restored file system to hold temporary archive files.

    In the example, we create the directory `/hqfs1/tars`

    ```
    root@mds1:~# mkdir /hqfs1/tars
    ```

11. Position the media at the beginning of the `tar` header for the archive file that holds the archived copy, and read the archive from the media into memory. Use the command `request -m` *`media-type`* `-v` *`volume-serial-number`* `-p 0x`*`position`* *`path`*`/`*`requestfile`*, where:

    - `-m` *`media-type`* species one of the two-character media type codes listed in Appendix B.
    - `-v` *`volume-serial-number`* specifies the six-character, alphanumeric code that identifies the media volume.
    - `-p 0x`*`position`* specifies the hexadecimal starting position that you noted in the archiver log entry.
    - *`path`* is the path to the temporary recovery directory.
    - *`requestfile`* is the name that you want to use for the in-memory `tar` file that the `request` command reads from the media.

    In the example, we create a request file, `/hqfs1/tars/currentrequest` starting from position `0x78` on LTO (`li`) volume `VOL012`:

    ```
    root@mds1:~# request -m li -v VOL012 -p 0x78 /hqfs1/tars/currentrequest
    ```

12. Extract the archive copy of the missing or damaged file from the in-memory `tar` file that you created in the preceding step. Use the command `star -xv -f` *`requestfile`*, where:

    - *`requestfile`* is the name of the in-memory `tar` file.
    - *`file`* is the path—relative to the file-system root directory—and name of the file that you need to restore.

    The star command is an enhanced Oracle HSM version of GNU `tar` that restores specified files from the request file (the in-memory copy of the archive file).

    In the example, we extract the data file `genfiles/genA0` from the request file `tars/currentrequest`. The file is restored to `/hqfs1/genfiles/genA0`:

    ```
    root@mds1:~# star -xvf tars/currentrequest genfiles/genA0
    ```

13. Set any required file attributes.

    When you restore a file from a `tar` file, without a `samfsdump` or `qfsdump` recovery point file, the original file attributes are lost. An `.inodes` file has to be created for the file from scratch, using default attribute values.

14. Repeat this procedure until all required files have been recovered.

15. If necessary, restore lost and damaged segmented files and/or volume overflow files.

16. Otherwise, finish the recovery procedure. Go to "Restoring Archiving File Systems to Normal Operation" on page 6-1.

Restore Lost and Damaged Segmented Files

Restoring a segmented file is much like restoring a regular file. However, you recover the individual segments rather than the file itself. So, to restore the file, you must reassemble the segments into a single file and then re-segment the result. For each file that you need to recover, proceed as follows:

1. If you have not already done so, log in to the file-system metadata server as `root`.

   ```
   root@mds1:~#
   ```

2. If you have not already done so, stop archiving and recycling using the procedure in "Stopping Archiving and Recycling Processes" on page 2-1

3. If you have an archiver log for the period when the segmented file was last archived, search the entries for segmented (type `S`) files. Select the most recent entries for segments of the required file.

   ```
   A 2016/10/24 14:01:47 li VOL013 all.1 76a.1 hqfs1 14.5 10485760 bf/dat011/1 S 0
   51
   A 2016/10/24 14:04:11 li VOL013 all.1 2476f.5002 hqfs1 15.5 10485760
   bf/dat011/2 S 0 51
   A 2016/10/24 14:06:24 li VOL013 all.1 1409aa4.1 hqfs1 16.5 184 bf/dat011/3 S 0
   51
   ```

4. Once you have located the latest entries for the segments, note the following details:

   - the media type
   - the volume serial numbers of the media volumes that store file segments
   - the hexadecimal starting positions of the archive (`tar`) files that hold the segments
   - the path and name of the segmented file relative to the root directory of the file system
   - the number of segments in the file.

   In the example, the file `dat011` is divided into three segments (`1`, `2`, and `3`). The three segments are stored in three archive files, all on a single LTO (`li`) tape volume, volume serial number `VOL013`. The three archive files start at positions 0x76a (`76a`), 0x2476f (`2476f`), and 0x1409aa4 (`1409aa4`)

   ```
   A 2016/10/24 14:01:47 li VOL013 all.1 76a.1 hqfs1 14.5 10485760 bf/dat011/1 S 0
   51
   A 2016/10/24 14:04:11 li VOL013 all.1 2476f.5002 hqfs1 15.5 10485760
   bf/dat011/2 S 0 51
   A 2016/10/24 14:06:24 li VOL013 all.1 1409aa4.1 hqfs1 16.5 184 bf/dat011/3 S 0
   51
   ```

5. Change to the root directory of the file system that you are restoring.

   Oracle HSM archive files store copies relative to the file-system root directory. So to restore them to their original locations, we want to restore them from the root directory.

   In the example, we change to the root of the `hqfs1` file system.

   ```
   root@mds1:~# cd /hsm/hqfs1
   ```

6. Create a directory in the restored file system to hold temporary archive files.

   In the example, we create the directory `/hsm/hqfs1/tars`

```
root@mds1:~# mkdir /hsm/hqfs1/tars
```

7. Position the media at beginning of each archive file that holds an archived copy of one or more of the file segments, and read the archive from the media into memory. Use the command `request -m` *`media-type`* `-v` *`volume-serial-number`* `-p 0x`*`position`* *`path`*`/`*`requestfile`*, where:

   - `-m` *`media-type`* specifies one of the two-character media type codes listed in Appendix B.
   - `-v` *`volume-serial-number`* specifies the six-character alphanumeric code that identifies the media volume.
   - `-p 0x`*`position`* specifies the hexadecimal starting position that you noted in the archiver log entry.
   - *`path`* is the path to the temporary recovery directory.
   - *`requestfile`* is the name that you want to use for the in-memory `tar` file that the `request` command reads from the media.

   In the example, we need to create two request files. The first, `/hqfs1/tars/request76a`, loads the archive file that starts at position `0x76a` on LTO (`li`) `VOL013`. This archive contains both of the first two segments. The second request file, `/hsm/hqfs1/tars/request1409aa4`, loads the archive file at position `0x1409aa4`, in this case on the same volume (segments can reside on any volume in the library):

   ```
   root@mds1:~# request -m li -v VOL013 -p 0x76a /hsm/hqfs1/tars/request76a
   root@mds1:~# request -m li -v VOL013 -p 0x1409aa4
   /hsm/hqfs1/tars/request1409aa4
   root@mds1:~#
   ```

8. Extract each segment of the backup copy of the missing or damaged file from the in-memory `tar` file that you created in the preceding step. Use the command `star -xv -f` *`requestfile`* *`segment`*, where *`requestfile`* is the name of the in-memory `tar` file and *`segment`* is the path—relative to the file-system root directory—and name of each segment of the file that you need to restore.

   The star command is an enhanced Oracle HSM version of GNU `tar` that restores specified files from the archive file that you are pointing to with the request file.

   In the example, we extract two of the three segments of the data file `bf/dat011` from the request file (in-memory `tar` files) `tars/request76a` and one from the request file `tars/request1409aa4`. The file is restored to a directory, `/hsm/hqfs1/bf/dat011/`, in three separate pieces:

   ```
   root@mds1:~# star -xvf tars/request76a bf/dat011/1
   root@mds1:~# star -xvf tars/request76a bf/dat011/2
   root@mds1:~# star -xvf tars/request1409aa4 bf/dat011/3
   ```

   When we list the contents of `/hsm/hqfs1/bf/dat011`, we see one sequentially numbered file for each restored segment:

   ```
   root@mds1:~# ls /hsm/hqfs1/bf/dat011
   total 40968
   -rw-rw---- 1 root other 10485760 Mar  5 17:06 1
   -rw-rw---- 1 root other 10485760 Mar  5 17:06 2
   -rw-rw---- 1 root other      184 Mar  5 17:07 3
   root@mds1:~#
   ```

9. Re-assemble the restored segments into a single, unsegmented, temporary file.

In the example, we concatenate the three segments in the `/hsm/hqfs1/bf/dat011/` directory to create the file `/hsm/hqfs1/bf/dat011file`:

```
root@mds1:~# cat /hsm/hqfs1/bf/dat011/1 /hsm/hqfs1/bf/dat011/2 \
/hsmfs/bf/dat011/3 > /hsm/hqfs1/bf/dat011file
root@mds1:~#
```

When we list the contents of `/hsm/hqfs1/bf/`, the new file appears alongside the directory containing the segments.

```
root@mds1:~# ls -l /hsm/hqfs1/bf/dat011*
drwxr-xr-x 2 root root      4096 Mar  5 17:06 dat011
-rw-rw---- 1 root other 20971704 Mar  5 17:14 dat011file
root@mds1:~#
```

10. Remove the segments and the directory that contains them.

```
root@mds1:~# rm -r /hsm/hqfs1/bf/dat011/
root@mds1:~#
```

11. Create an empty file using the original path and name of the segmented file. Use the command `touch` *`file`*, where *`file`* is the original path and file name.

    In the example, we create the empty file `/hsm/hqfs1/bf/dat011`, the original name of the segmented file that we are restoring:

```
root@mds1:~# touch /hsm/hqfs1/bf/dat011
root@mds1:~#
```

12. Set the Oracle HSM segment attribute on the newly created, empty file. Use the command `segment -l` *`segment-length file`*, where *`segment-length`* is the segment length that you noted in the archiver log entry and *`file`* is the original path and name of the segmented file.

    In the example, the archiver log shows that the segment length for the file `dat011` is `10485760` (the file ends in the third segment, so the length of the data on the media is less than the segment length):

```
A 2016/10/24 14:01:47 li VOL013 all.1 76a.1 hqfs1 14.5 10485760 bf/dat011/1 S 0
51
A 2016/10/24 14:04:11 li VOL013 all.1 76a.5002 hqfs1 15.5 10485760 bf/dat011/2
S 0 51
A 2016/10/24 14:06:24 li VOL013 all.1 1409aa4.1 hqfs1 16.5 184 bf/dat011/3 S 0
51
```

So we set the segment length for the empty file to `10485760`:

```
root@mds1:~# segment -l 10485760 /hsm/hqfs1/bf/dat011
root@mds1:~#
```

13. Copy the unsegmented temporary file to the empty segmented file.

    In the example, we copy `dat011file` to `dat011`:

```
root@mds1:~# cp /hsm/hqfs1/bf/dat011file /hsm/hqfs1/bf/dat011
root@mds1:~#
```

When we use the command `sls -2K` `hsm/hqfs1/bf/dat011` to list the segments, they are listed as shown. So the file has been restored.

```
root@mds1:~# sls -2K /hsm/hqfs1/bf/dat011
-rw-rw---- 1 root other       20971704     Mar  5 17:12 /hsm/hqfs1/bf/dat011
---------- ----- sI {3,0,0,0}
-rw-rw---- 1 root other       10485760     Mar  5 17:12 /hsm/hqfs1/bf/dat011/1
```

```
---------- ----- sS
-rw-rw---- 1 root other       10485760     Mar  5 17:12 /hsm/hqfs1/bf/dat011/2
---------- ----- sS
-rw-rw---- 1 root other            184     Mar  5 17:12 /hsm/hqfs1/bf/dat011/3
---------- ----- sS
```

14. Set any other required file attributes.

    When you restore a file from a `tar` file, without a `samfsdump` or `qfsdump` recovery point file, the original file attributes are lost. An `.inodes` file has to be created for the file from scratch, using default attribute values.

15. The file has now been restored. Delete the unsegmented temporary file.

    In the example, we delete `dat011file`:

    ```
    root@mds1:~# rm /hsm/hqfs1/bf/dat011file
    root@mds1:~#
    ```

16. Repeat this procedure until all required files have been recovered.

17. Finish the recovery procedure. Go to "Restoring Archiving File Systems to Normal Operation" on page 6-1.

Restore Lost and Damaged Volume Overflow Files

A volume overflow file is a regular file that spans media volumes. Restoring a volume overflow file is thus much like restoring any other regular file. However, you must combine sections of an archive file that resides on multiple volumes into a single archive file on disk before you extract the data file from the archive. So, for each file that you need to recover, proceed as follows:

1. If you have not already done so, log in to the file-system metadata server as `root`.

    ```
    root@mds1:~#
    ```

2. If you have not already done so, stop archiving and recycling using the procedure in "Stopping Archiving and Recycling Processes" on page 2-1

3. If you have an archiver log for the period when the volume overflow file was last archived, find the most recent entry for the file. Note the volume serial number(s) of the media, the length of each section of the file, the path and name of the file relative to the root directory of the file system, and the number of sections in the file.

    In the example, we know that the file `/hsm/hqfs1/rf/rf81` is a volume overflow because it is a regular, type `f` file that resides on two volumes, `VOL036` and `VOL034`, and has two sections, `0` and `1`:

    ```
    A 2016/10/24 18:28:51 li VOL036 all.1 12d.1 hqfs1 11731.1 89128448  rf/rf81 f 0
    210
    A 2016/10/24 18:28:51 li VOL034 all.1 15f.0 hqfs1 11731.1 525271552 rf/rf81 f 1
    220
    ```

4. Change to the root directory of the file system that you are restoring.

    Oracle HSM archive files store copies relative to the file-system root directory. So to restore them to their original locations, we want to restore from to the root directory.

    In the example, we change to the root of the `hqfs1` file system.

    ```
    root@mds1:~# cd /hsm/hqfs1
    ```

5. Before proceeding, make sure that the file system contains enough free space accommodate a file at least twice the size of the file that you are recovering.

   For the file in the example, `rf/rf81`, we will need about 1.2 gigabytes of free space, based on the sizes of the two sections of the file: 2 x (89128448 + 525271552) = 1228800000 bytes.

6. Create a directory in the restored file system to hold temporary archive files.

   In the example, we create the directory `/hsm/hqfs1/tars`

   ```
   root@mds1:~# mkdir /hsm/hqfs1/tars
   ```

7. Position the media at beginning of each archive file that holds an archived copy of one or more of the file segments, and read the archive from the media into memory. Use the command `request -m` *`media-type`* `-v` *`volume-serial-number`* `-p 0x`*`position path`*`/`*`requestfile`*, where:

   - `-m` *`media-type`* specifies one of the two-character media type codes listed in Appendix B.
   - `-v` *`volume-serial-number`* specifies the six-character alphanumeric code that identifies the media volume.
   - `-p 0x`*`position`* is the hexadecimal starting position that you noted in the archiver log entry.
   - *`path`* is the path to the temporary recovery directory.
   - *`requestfile`* is the name that you want to use for the in-memory `tar` file that the `request` command reads from the media.

   In the example, we create two request files. The first request file, `/hqfs1/tars/requestVOL036`, loads the archive file that starts at position `0x12d` on LTO (`li`) `VOL036`. The second request file, `/hqfs1/tars/requestVOL034`, loads the archive file at position `0x15f` on LTO (`li`) `VOL034`:

   ```
   root@mds1:~# request -m li -v VOL036 -p 0x12d /hsm/hqfs1/tars/requestVOL036
   root@mds1:~# request -m li -v VOL034 -p 0x15f /hsm/hqfs1/tars/requestVOL034
   ```

8. Save each of the in-memory `tar` files that you created to disk as a section of the archive file. Use the command `dd if=` *`requestfile`* `of=`*`archive_section`*, where *`requestfile`* is the path and name of the in-memory `tar` file and *`archive_section`* is the path and name of each section of the archive file.

   In the example, we save the request files (in-memory `tar` files), `tars/requestVOL036` and `tars/requestVOL034` as `tars/archive_part1` and `tars/archive_part2`:

   ```
   root@mds1:~# dd if=tars/requestVOL036 of=tars/archive_part1
   root@mds1:~# dd if=tars/requestVOL034 of=tars/archive_part2
   root@mds1:~#
   ```

9. Re-assemble the sections into a single archive file.

   In the example, we concatenate the two sections, `tars/archive_part1` and `tars/archive_part2`, to create a single archive file, `/tars/archive_complete`:

   ```
   root@mds1:~# cat tars/archive_part1 tars/archive_part2 > tars/archive_complete
   root@mds1:~#
   ```

10. Extract the backup copy of the missing or damaged volume overflow file from the archive (`tar`) file that you created in the preceding step. Use the command `star`

-xv -f *tarfile file*, where *tarfile* is the name of the archive file and *file* is the path—relative to the file-system root directory—and name of the volume overflow file that you need to restore.

The star command is an enhanced Oracle HSM version of GNU tar that restores specified files from the archive file that you are pointing to with the request file.

In the example, we extract the volume overflow file rf/rf81 from the tar file tars/archive_complete:

```
root@mds1:~# star -xvf tars/archive_complete rf/rf81
```

11. Set any other required file attributes.

    When you restore a file from a tar file, without a samfsdump or qfsdump recovery point file, the original file attributes are lost. An .inodes file has to be created for the file from scratch, using default attribute values.

12. The volume overflow file has now been restored. Delete the temporary file.

    In the example, we delete archive_complete:

    ```
    root@mds1:~# rm tars/archive_complete
    root@mds1:~#
    ```

13. Repeat this procedure until all required files have been recovered.

14. When all required files have been recovered, delete the temporary directory.

    In the example, we delete the directory /hsm/hqfs1/tars:

    ```
    root@mds1:~# rm -R tars/
    root@mds1:~#
    ```

15. Finish the recovery procedure. Go to "Restoring Archiving File Systems to Normal Operation" on page 6-1.

Recovering Damaged Archive Copies

A *damaged* archive copy is a copy of a file that cannot be staged back to the disk cache. Sometimes, the file has merely failed to stage due to an intermittent, hardware-related I/O problem and can be easily resolved. At other times, the damaged copy is corrupt and the data is irrecoverable. Your only option in such cases is to recover the file from an alternate copy.

To identify and manage damaged copies, proceed as follows:

1. Identify files that have damaged archive copies. Use the command sfind *mountpoint* -any_copy_d, where *mountpoint* is the directory where the recovered file system is mounted.

   In the example, we start the search in the directory /hsm/hqfs1 and find three files that have damaged copies:

   ```
   root@mds1:~# sfind /hsm/hqfs1 -any_copy_d
   ./genfiles/ab09
   ./genfiles/ab11
   ./genfiles/ay12
   root@mds1:~#
   ```

2. For each file that you identified, identify the damaged copies. Use the command sls -D *file*, where *file* is the path and file name that you want to check.

Damaged copies are flagged with a `D`. In the example, **`copy 2`** of **`/hsm/hqfs1/genfiles/ab09`** and **`copy 1`** of **`/hsm/hqfs1/genfiles/ab11`** are damaged:

```
root@mds1:~# sls -D /hsm/hqfs1/genfiles/ab09
/hsm/hqfs1/genfiles/ab09:
  mode: -rw-r-----  links:   1  owner: root group: other
  length:    306581  admin id: 0  inode:    11748.11
  project: system(0)
  copy 1: ---- Mar 11 13:52       76f.421bc li VOL011
  copy 2: ---D Mar 31 14:02       286.1324f li VOL021
  access:   Mar 11 13:50  modification: Mar 11 13:50
  changed:  Mar 11 13:50  attributes:   Mar 11 13:50
  creation: Mar 11 13:50  residence:    Mar 11 13:50
root@mds1:~# sls -D /hsm/hqfs1/genfiles/ab11
/hsm/hqfs1/genfiles/ab11:
  mode: -rw-r-----  links:   1  owner: root group: other
  length:    380051  admin id: 0  inode:    1460.1
  project: system(0)
  copy 1: ---D Mar 01 10:21       431.21bc6 li VOL024
  access:   Mar 01 10:21  modification: Mar 01 10:21
  changed:  Mar 01 10:21  attributes:   Mar 01 10:21
  creation: Mar 01 10:21  residence:    Mar 01 10:21
root@mds1:~#
```

3. If there is an alternate copy, unarchive the damaged copy. Use the command `unarchive -c` *`CopyNumber file`*, where *`CopyNumber`* is an integer representing the copy number and *`file`* is the path and file name of the damaged file. Stop here.

   When you unarchive the damaged copy, Oracle HSM stages from the remaining copy and creates an additional archive copy the next time that the archiver process runs. In the example, we have another, undamaged copy of `/hsm/hqfs1/genfiles/ab09`, so we unarchive the damaged copy, copy `2`:

   ```
   root@mds1:~# unarchive -c 2 /hsm/hqfs1/genfiles/ab09
   root@mds1:~#
   ```

4. If you do not have another copy, undamage the damaged copy. Use the command `undamage -c`*`CopyNumber file`*, where *`CopyNumber`* is an integer representing the copy number and *`file`* is the path and file name of the damaged file.

   Sometimes a file fails to stage due to an intermittent, hardware-related I/O error. Clearing the damage flag and restaging may solve the problem. In the example, there is only one copy of `/hsm/hqfs1/genfiles/ab11`:

   ```
   root@mds1:~# undamage -c1 /hsm/hqfs1/genfiles/ab11
   ```

5. Try to stage the copy. Use the command `stage -c` *`CopyNumber`* `-I` *`file`*, where *`CopyNumber`* is an integer representing the copy number and *`file`* is the path and file name of the file.

   The optional `-I` (immediate) parameter pushes the staging operation to the head of the queue:

   ```
   root@mds1:~# stage -c 1 -I /hsm/hqfs1/genfiles/ab11
   ```

6. If staging succeeds, stop here.

7. If staging failed again, Oracle HSM again sets the damaged flag. Note the major inode number in the output of the `sls -D` command for the damaged copy.

   In the example, the inode number of the file, `/hsm/hqfs1/genfiles/ab11`, is `1460`:

```
root@mds1:~# sls -D /hsm/hqfs1/genfiles/ab11
/hsm/hqfs1/genfiles/ab11:
  mode: -rw-r-----  links:   1  owner: root group: other
  length:    380051  admin id: 0  inode:    1460.1
  project: system(0)
  copy 1: ---D Mar 01 10:21       431.21bc6 li VOL024
  ...
root@mds1:~#
```

8. Look for possible causes. First, examine the Oracle HSM `/var/adm/sam-log` file for staging related messages that pertain to the inode of the file with the damaged copy.

   The search can be carried out in various ways. In the example, we list the contents of log file using the Solaris `cat` command and pipe the output to `grep` and a regular expression that matches the inode number. We find two messages. Both indicate an I/O error and one explicitly implicates equipment (`eq`) ordinal number `804`, one of our tape drives:

```
root@mds1:~# cat /var/adm/sam-log | grep "inode 1460"
Mar 11 15:35:44 server1 genu-20[8899]: Stage request canceled for inode 1460
(eq 804): I/O error.
Jan 11 15:35:44 server1 samfs[8894]: /sam4 inode 1460.1 - Archive copy 1 marked
damaged: I/O error
```

9. If the `/var/adm/sam-log` file implicates a specific Oracle HSM equipment ordinal number, examine the device log, `/var/opt/SUNWsamfs/devlog/`*`drive-equipment-number`*, where *`drive-equipment-number`* is the ordinal number listed in the `/var/adm/sam-log` file.

10. If the problem appears to be specific to a particular drive, make the implicated drive unavailable to the staging process using the command `samcmd unavail` *`drive-equipment-number`*. Then undamage the copy, and try to stage it again.

```
root@mds1:~# samcmd unavail 804
root@mds1:~# stage -c 1 -I /hsm/hqfs1/genfiles/ab11
root@mds1:~# undamage -c 1 /hsm/hqfs1/genfiles/ab11
root@mds1:~#
```

11. If staging fails again or if no single drive appears to be at fault, try to recover the copy using the `request` and `star` commands, as described in "Recovering Files Using Log Entries" on page 5-3, or Solaris utilities such as `tar` and `dd`.

12. If you still cannot recover the file and if the value of the data warrants it, engage a data recovery service. For assistance with Oracle StorageTek tape media, engage Oracle StorageTek Enterprise Tape Data Recovery services. Log in to My Oracle Support at `support.oracle.com`. Open a Service Request, select the tape drive model from the list under the request category, and select Media Issues from the list under subcategory.

13. If the file proves to be irrecoverable, unarchive the damaged copy. Use the command `unarchive -c` *`CopyNumber file`*, where *`CopyNumber`* is an integer representing the copy number and *`file`* is the path and file name of the damaged file.

```
root@mds1:~# unarchive -c 1 /hsm/hqfs1/genfiles/ab11
root@mds1:~#
```

14. Resolve any drive or media issues that are revealed by the log files.

15. If you disabled archiving, staging, and recycling processes in a previous step, re-enable them now. Go to "Restoring Archiving File Systems to Normal Operation" on page 6-1.

16. Otherwise, stop here.

6

Finishing Up

Once you have completed configuration and data recovery, you should perform two last tasks:

- restoring archiving file systems to normal operation
- preserving the new configuration information.

Restoring Archiving File Systems to Normal Operation

If you disabled archiving and recycling, re-enable them now:

- enable archiving
- enable recycling.

Enable Archiving

1. Log in to the file-system metadata server as `root`.

   ```
   root@mds1:~#
   ```

2. Open the `/etc/opt/SUNWsamfs/archiver.cmd` file in a text editor, and scroll down to the first `wait` directive that you added to the file when you started recovery efforts.

   In the example, we use the `vi` editor:

   ```
   root@mds1:~# vi /etc/opt/SUNWsamfs/archiver.cmd
   # Configuration file for Oracle HSM archiving file systems
   #-----------------------------------------------------------------------
   # General Directives
   archivemeta = off
   examine = noscan
   #-----------------------------------------------------------------------
   # Archive Set Assignments
   wait
   fs = hsmfs1
   logfile = /var/adm/hsmfs1.archive.log
   all .
       1 -norelease 15m
       2 -norelease 15m
   fs = hsmfs2
   logfile = /var/adm/hsmfs2.archive.log
   all .
   ...
   ```

3. To enable archiving, delete every `wait` directive that you added to the file when you started recovery efforts. Save the file, and close the editor.

   In the example, we remove the single `wait` directive that we added:

```
root@mds1:~# vi /etc/opt/SUNWsamfs/archiver.cmd
...
#-----------------------------------------------------------------------
# Archive Set Assignments
fs = hsmfs1
logfile = /var/adm/hsmfs1.archive.log
all .
    1 -norelease 15m
    2 -norelease 15m
    3 -norelease 15m
fs = hsmfs2
...
:wq
root@mds1:~#
```

4. Next, enable recycling.

Enable Recycling

1. Log in to the file-system metadata server as `root`.

```
root@mds1:~#
```

2. Open the `/etc/opt/SUNWsamfs/recycler.cmd` file in a text editor, and scroll down to the first `-ignore` parameter that you added to the file when you started recovery efforts.

   In the example, we use the `vi` editor:

```
root@mds1:~# vi /etc/opt/SUNWsamfs/recycler.cmd
# Configuration file for Oracle HSM archiving file systems
#-----------------------------------------------------------------------
logfile = /var/adm/recycler.log
no_recycle tp VOL[0-9][2-9][0-9]
lib1 -hwm 95 -mingain 60 -ignore
```

3. Remove every `-ignore` parameter that you added when you started recovery efforts. Then save the file, and close the editor.

   In the example, we have only one `-ignore` parameter in the Oracle HSM configuration:

```
root@mds1:~# vi /etc/opt/SUNWsamfs/recycler.cmd
# Configuration file for Oracle HSM archiving file systems
#-----------------------------------------------------------------------
logfile = /var/adm/recycler.log
no_recycle tp VOL[0-9][2-9][0-9]
lib1 -hwm 95 -mingain 60
:wq
root@mds1:~#
```

4. Check the modified configuration files for errors. Use the commands `archiver -lv` to check the `archiver.cmd` file, and run the initialization command `sam-fsd`. Correct any errors.

   In the example, the configuration files are correct:

```
root@mds1:~# archiver -lv
```

```
Reading '/etc/opt/SUNWsamfs/archiver.cmd'.
1: #-----------------------------------------------------------------------
2: # General Directives
3: archivemeta = off
4: examine = noscan
5: #-----------------------------------------------------------------------
5: # Archive Set Assignments
7: fs = hsmfs1
...
   .sort: path
root@mds1:~# sam-fsd
Trace file controls:
sam-amld      /var/opt/SUNWsamfs/trace/sam-amld
...
Would start sam-archiverd()
Would start sam-stagealld()
Would start sam-stagerd()
Would start sam-amld()
root@mds1:~#
```

5. Reconfigure the Oracle HSM software using the restored configuration files. Use the command `samd config`.

   Archiving and recycling processes resume.

   ```
   root@mds1:~# samd config
   ```

6. If you are recovering from a server problem or from loss or damage to one more file systems, save the newly restored Oracle HSM configuration.
7. Otherwise, stop here.

Preserving the New Configuration Information

If you have changed the Oracle HSM configuration in the course of recovery efforts, you should back up the configuration again now.

Save the Newly Restored Oracle HSM Configuration

1. Log in to the file-system metadata server as `root`.

   ```
   root@mds1:~#
   ```

2. Run the `samexplorer` command and create a SAMreport. Save it in the directory that holds your backup configuration information. Use the command `samexplorer` *`path`*`/`*`hostname`*`.`*`YYYYMMDD`*`.`*`hhmmz`*`.tar.gz`, where *`path`* is the path to the chosen directory, *`hostname`* is the name of the Oracle HSM file system host, and *`YYYYMMDD`*`.`*`hhmmz`* is a date and time stamp.

   The default file name is `/tmp/SAMreport.`*`hostname`***`.`***`YYYYMMDD`*`.`*`hhmmz`*`.tar.gz`. In the example, we already have a directory for saving SAMreports, `/zfs1/sam_config/`. So we create the report in this directory:

   ```
   root@mds1:~# samexplorer
   /zfs1/sam_config/explorer/server1.20140430.1659MST.tar.gz
        Report name:
   /zfs1/sam_config/explorer/samhost1.20140430.1659MST.tar.gz
        Lines per file:  1000
        Output format:   tar.gz (default) Use -u for unarchived/uncompressed.

        Please wait............................................
   ```

```
Please wait..........................................
Please wait....................................

The following files should now be ftp'ed to your support provider
as ftp type binary.

/zfs1/sam_config/explorer/samhost1.20140430.1659MST.tar.gz
```

3. Copy the `/etc/opt/SUNWsamfs/` directory and its contents to an independent file system.

   The `/etc/opt/SUNWsamfs/` directory may contain any or all of the following:

   - `mcf` (the master configuration file for the Oracle HSM file systems)
   - `archiver.cmd` (configures the archiving process)
   - `inquiry.conf` (lists the vendor and product identification strings that SCSI devices report in response to an inquiry command)
   - `scripts/*` (locally customized Oracle HSM scripts)
   - `defaults.conf` (overrides specified, default parameter values)
   - `diskvols.conf` (identifies disk storage that is used for archiving)
   - `hosts.`*`family-set-name`* (defines server and client host names and IP addresses for a shared file-system)
   - `hosts.`*`family-set-name`*`.local` (defines server and client host names and IP addresses for a shared file-system)
   - `preview.cmd` (customizes the priorities of archiving and staging requests for volumes that are not currently loaded)
   - `recycler.cmd` (customizes the recycling process)
   - `releaser.cmd` (customizes the releasing process)
   - `rft.cmd` (controls the Oracle HSM file transfer service)
   - `samfs.cmd` (defines file system mount parameters)
   - `stager.cmd` (customizes the staging process)
   - `samremote` (the SAM-Remote server configuration file)
   - *`family-set-name`* (a SAM-Remote client configuration file)
   - *`network-attached-library`* (a parameters file for a network-attached library

4. Back up the library catalogs, including the historian catalog. For each catalog, use the command `dump_cat -V` *`catalog-file`*, where *`catalog-file`* is the path and name of the catalog file. Redirect the output to *`dump-file`* in an independent file system.

   In the example, we first dump the catalog data for `lib1` to the file `lib1cat2.dump` in a directory on the independent NFS-mounted file system `zfs1`. Then we dump the historian catalog:

```
root@mds1:~# dump_cat -V /var/opt/SUNWsamfs/catalog/lib1 >
/zfs1/hsmcfg/lib1cat2.dump
root@mds1:~# dump_cat -V /var/opt/SUNWsamfs/catalog/historian >
/zfs1/hsmcfg/historian2.dump
```

5. Copy system configuration files that were modified during Oracle HSM installation and configuration. These may include:

```
/etc/
     syslog.conf
     system
     vfstab
/kernel/drv/
     sgen.conf
     samst.conf
     samrd.conf
     sd.conf
     ssd.conf
     st.conf
/usr/kernel/drv/dst.conf
```

6. Copy any custom shell scripts and `crontab` entries that you created as part of the Oracle HSM configuration to the selected subdirectory.

   For example, if you created a `crontab` entry to manage creation of recovery points, you would save a copy now.

7. Record the revision level of the currently installed software, including Oracle Oracle HSM, Solaris, and Solaris Cluster (if applicable), and save a copy of the information in a `readme` file in the chosen subdirectory.

8. In the chosen subdirectory, save copies of any newly downloaded Oracle Oracle HSM, Solaris, and Solaris Cluster packages so that you can restore the software quickly, should it again become necessary.

A

Understanding Archiver and Migration Logs

Archiver and migration logs record the exact locations of files that have been copied to tape. Should you need to recover a file system, these log files contain information that lets you restore any files that cannot be found using the available recovery point files.

The following table defines each field in the archiver log.

Field	Typical Value	Meaning
1	`A`	The type of archive activity logged: `A` (*archived*), `R` (*re-archived*), or `U` (*unarchived*)
2	`2015/03/23`	The date of the archive action, in the form *yyyy*/*mm*/*dd*.
3	`18:42:06`	The time of the archive activity, in the form *hh*:*mm*:*ss*.
4	`ti`	The archive media type. Media types, are discussed in Appendix B, "Glossary of Equipment Types" and on the `mcf(4)` man page.
5	`VOL004`	The volume serial number (VSN) of a removable media volume or the volume name and `tar`(1) path to a file archived on disk media.
6	`arset0.1`	The Oracle HSM archive set name and copy number.
7	`9a089.132`	The physical position of the start of the archive file (tar file) on media and the file offset within the archive file, in hexadecimal format.
8	`hsm1`	The name of the file system that holds the file.
9	`118.51`	The inode number and generation number of the file. The generation number is used in addition to the inode number for uniqueness because inode numbers are reused.
10	`162514`	The 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`	The path and name of the file relative to the mount point of the file system.
12	`f`	The type of file: `d` (*directory*), `f` (*file*), `l` (*symbolic link*), `R` (*removable-media file*), `I` (*segment index*), or `S` (*data segment*)
13	`0`	The section number of an overflowed file or segment. If the file is an overflowed file, the value is non-zero. Otherwise the value is 0.
14	`56`	The equipment ordinal of the drive on which the file was archived.

The following example shows sample lines from an archiver log file.

```
A 2014/03/23 18:42:06 ti VOL004 arset0.1 9a089.1329 hsm1 118.51 162514 t0/fdn f 0 54
A 2014/03/23 18:42:10 ti VOL004 arset0.1 9aac2.1 hsm1 189.53 1515016 t0/fae f 0 56
A 2014/03/23 18:42:10 ti VOL004 arset0.1 9aac2.b92 hsm1 125.53 867101 t0/fai f 0 50
```

```
A 2014/03/24 13:30:24 dk DISK01/d8/d16/f2 arset4.1 810d8.1 hsm1 11971.30 1136048 t1/dat0 f 0 0
A 2014/03/24 13:30:25 dk DISK01/d8/d16/f2 arset4.1 810d8.8d hsm1 11973.9 1849474 t1/dat9 f 0 0
A 2014/03/24 13:30:25 dk DISK01/d8/d16/f3 arset4.1 810d8.96 hsm1 119576.6 644930 t1/file7 f 0 0
```

Media migration logs are very similar to archive logs. The archive-activity and the archive-set/copy-number fields are omitted, and the last three fields are unique to migration logs. All other fields are the same. Note, however, that there is a separate migration log for each volume migrated. Each log file is in the logging directory specified by the `migrationd.cmd` file.

Field	Typical Value	Meaning
1	`2015/03/23`	The date of the migration action, in the form *yyyy/mm/dd*.
2	`18:42:06`	The time of the migration activity, in the form *hh*:*mm*:*ss*.
3	`ti`	The archive media type. Media types, are discussed in Appendix B, "Glossary of Equipment Types" and on the `mcf(4)` man page.
4	`VOL004`	The volume serial number (VSN) of a removable media volume or the volume name and `tar`(1) path to a file archived on disk media.
5	`9a089.19`	The physical position of the start of the archive file (tar file) on media and the file offset within the archive file, in hexadecimal format.
6	`hsm1`	The name of the file system that holds the file.
7	`118.51`	The inode number and generation number of the file. The generation number is used in addition to the inode number for uniqueness because inode numbers are reused.
8	`162514`	The 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.
9	`dat0/datA`	The path and name of the file relative to the mount point of the file system.
10	`f`	The type of file: `d` (*directory*), `f` (*file*), `l` (*symbolic link*), `I` (*segment index*), or `S` (*data segment*)
11	`s`	The copy mode used for the migration: either `s` (*server copy*) or `x` (*xcopy*).
12	`801`	The equipment ordinal of the drive that mounted the source volume.
13	`804`	The equipment ordinal of the drive that mounted the destination volume.

The following example shows representative lines from the migration log file `hsm_migration_logs/li.VOL001`, which logs migration of the LTO tape `VOL001` to new media:

```
2015/10/16 12:14:12 li VOL012 2 4.1 hsmfs1 1026.1 0 .domain f s 804 801
2015/10/16 12:14:12 li VOL012 2 4.2 hsmfs1 1025.1 0 .fuid f s 804 801
2015/10/16 12:14:12 li VOL012 2 6.1 hsmfs1 1040.1 14971 data0/dat0A f s 804 801
2015/10/16 12:14:12 li VOL012 2 6.20 hsmfs1 1041.1 14971 data0/dat0B f s 804 801
2015/10/16 12:14:12 li VOL012 2 6.3f hsmfs1 1042.1 14971 data0/dat0C f s 804 801
2015/10/16 12:14:12 li VOL012 2 6.5e hsmfs1 1043.1 14971 data0/dat0D f s 804 801
```

B

Glossary of Equipment Types

The value of the `Equipment Type` field of the Master Configuration File (`mcf`) identifies devices and device configurations within the Oracle Hierarchical Storage Manager and StorageTek QFS Software. Equipment types are specified as two-character codes. This glossary lists the codes for quick reference when working with the samples or when interpreting an existing `mcf` (for full details see the `mcf(4)` man page).

For convenience, the codes are divided into two sections and then listed alphabetically:

- Recommended Equipment and Media Types
- Other Equipment and Media Types

Recommended Equipment and Media Types

This section describes all of the equipment codes that you normally need: the generic equipment codes (`rb`, `tp`, and `od`) and codes for identifying network-attached library interfaces, archival disk volumes, cloud resources, and the Oracle HSM historian.

The generic equipment codes `rb`, `tp`, and `od` are the preferred equipment type codes for all SCSI-attached libraries, tape drives, and optical disk devices. When you specify a generic equipment type, Oracle HSM can automatically set the correct type based on SCSI vendor codes.

cl

The abstract media type that organizes cloud resources into logical media volumes suitable for archiving. Oracle HSM labels each volume with a 31-character volume serial number (VSN) of the form *`nameNumber`*, where:

- *`name`* is the customer-defined string of 4 to 20 alphanumeric characters that identifies the logical library (type `cr`) that owns the media.
- *`Number`* is a randomly generated number in the range `[0000000-9999999]`.

cr

The abstract equipment type that manages cloud media (volumes of type `cl`) by emulating a network-attached tape library with a configurable number of drives.

A parameter file defines the characteristics of the equipment, including the `name` parameter value that prefixes and uniquely identifies the type `cl` volumes that belong to the library. See the `cloud` (7) and `sam-cloudd` (1m) man pages.

dk

A disk-based file system that the Oracle HSM software uses as an archival volume. UFS, ZFS, QFS, and NFS file systems can serve as disk archives.

Disk volumes and volume serial numbers (VSNs) are defined in the `/etc/opt/SUNWsamfs/diskvols.conf` file. See the `diskvols.conf` (4) man page.

g*XXX*
Where *`XXX`* is an integer in the range `[0-127]`, a striped group of disk devices that is part of an `ma` disk-cache family set.

hy
The Oracle HSM historian, an optional, virtual library that maintains a media catalog, but has no associated hardware. Used for tracking exported media.

ma
A high-performance QFS file system that maintains file-system metadata on one or more dedicated `mm` disk devices. File data resides on separate `md`, `mr`, or `g`*`XXX`* data devices.

md
A disk device that stores file data for an `ma` file system or data and metadata for an `ms` file system. `md` devices store file data in small, 4-kilobyte Disk Allocation Units (DAUs) and large, 16-, 32-, or 64-kilobyte DAUs. The default DAU is 64-kilobytes.

mm
A disk device that stores file-system metadata for a high-performance `ma` file system.

mr
A disk device that stores file data for an `ma` file system. `mr` devices store file data in large, fully adjustable Disk Allocation Units (DAUs) that are multiples of 8 kilobytes in the range 8-65528 kilobytes. The default DAU is 64 kilobytes.

ms
A Oracle HSM file system that maintains file-system metadata on the same devices that store file data.

od
Any SCSI-attached optical disk. Oracle HSM sets the appropriate equipment type automatically using the SCSI vendor code.

of
An abstract media type that distinguishes foreign media from Oracle HSM file system media. Used when migrating files from a foreign file system to an Oracle HSM file system.

The `of` media type does not identify a physical equipment type. Never specify it in a master configuration file (`mcf`).

rb
Any SCSI-attached tape library. Oracle HSM sets the appropriate equipment type automatically using the SCSI vendor code.

rd
The SAM-Remote pseudo-device. In the Master Configuration File (`mcf`), the corresponding `Equipment Identifier` field has to contain the path to the pseudo-device (such as `/dev/samrd/rd2`). The corresponding `Family Set` field has to contain the hostname of the SAM-Remote server.

sc
A SAM-Remote client system. In the Master Configuration File (`mcf`), the corresponding `Equipment Identifier` field has to contain the path the SAM-Remote

client-configuration file for the client. The corresponding `Family Set` field has to contain the family set name of the server. The `Additional Parameters` field must contain the full path to the client's library catalog file.

sk
An Oracle StorageTek ACSLS interface to a network-attached library. In the Master Configuration File (`mcf`), the corresponding `Equipment Identifier` field has to contain the path to the parameters file for the ACSLS interface. For more information, see the `stk(7)` man page.

ss
A SAM-Remote server. In the Master Configuration File (`mcf`), the corresponding `Equipment Identifier` field has to contain the path to the SAM-Remote server-configuration file. The corresponding `Family Set` field has to contain the family set name of the server, which must match the name used in the `Family Set` field of the `mcf` on the client.

tf
An abstract media type that distinguishes Oracle StorageTek T10000 and LTO volumes that are in Linear Tape File System (LTFS) format from volumes that contain Oracle HSM archive files.

The `tf` media type does not identify a physical equipment type. Never specify it in a master configuration file (`mcf`).

tp
Any SCSI-attached tape drive. Oracle HSM sets the appropriate equipment type automatically using the SCSI vendor code. No, however, that if you do use more specific equipment codes such as `li` and `ti`, you must do so consistently. If you specify `li` (LTO) tape equipment in the `mcf` file, for example, you cannot refer to the same equipment as `tp` equipment in the `archiver.cmd` file

z*X* (where *X* is a character in the range [0-9a-z])
An abstract media type that distinguishes foreign media from media controlled by Oracle HSM software.

Other Equipment and Media Types

The equipment types listed in this section are also supported.

Note that, in most cases, Oracle recommends identifying SCSI-attached libraries, tape drives, and optical disk devices using the generic equipment types `rb`, `tp`, and `od`. The generic equipment types tell Oracle HSM to identify the hardware dynamically, using SCSI vendor IDs. The type codes below are essential when migrating from one media type to another and may sometimes be useful for management purposes. But using them in a Master Configuration File (`mcf`), for example, hard-codes a static equipment configuration that may, at some point, no longer match the actual hardware.

ac
A Sun 1800, 3500, or L11000 tape library.

at
A Sony AIT-4 or AIT-5 tape drive.

cy
A Cygnet optical disk library.

d3
A StorageTek D3 tape drive.

dm
A Sony DMF library.

ds
A DocuStore or Plasmon optical disk library.

dt
A DAT 4-mm tape drive.

e8
An Exabyte X80 library.

fd
A Fujitsu M8100 128-track tape drive.

h4
An HP SL48 or SL24 library.

hc
An Hewlett Packard L9-/L20-/L60-series library.

i7
An IBM 3570 tape drive.

ic
An IBM 3570 media changer.

il
An IBM 3584 tape library.

li
An LTO-3 or later tape drive.

lt
A Digital Linear Tape (DLT), Super DLT, or DLT-S4 tape drive.

me
A Metrum library.

mf
An IBM Multi Function optical drive.

mo
A 5.25-in erasable optical drive.

o2
A 12-in WORM drive.

ov
An Overland Data Inc. Neo Series tape library.

pd
A Plasmon D-Series DVD-RAM library.

q8
A Qualstar 42xx, 62xx, or 82xx library.

s3
A StorageTek SL3000 library.

s9
An Oracle StorageTek 97xx series library.

se
A StorageTek 9490 tape drive.

sf
A StorageTek T9940 tape drive.

sg
A StorageTek 9840C or later tape drive.

sl
A Spectra Logic or Qualstar tape library.

st
A StorageTek 3480 tape drive.

ti
A StorageTek T10000 (Titanium) tape drive.

vt
A Metrum VHS (RSP-2150) tape drive.

wo
A 5.25-in optical WORM drive.

xt
An Exabyte (850x) 8-mm tape drive.

C

Product Accessibility Features

Users with low vision, blindness, color blindness, or other visual impairments can access the Oracle Hierarchical Storage Manager and StorageTek QFS Software (Oracle HSM) via the commandline interface. This text-based interface is compatible with screen readers, and all functions are controlled using a keyboard.

Glossary

This glossary focuses on terms specific to Oracle Hierarchical Storage Manager and StorageTek QFS Software and file systems. For industry standard definitions, please refer to the dictionary maintained by the Storage Networking Industry Association at `http://www.snia.org/education/dictionary/`.

active metadata server

See **metadata server (MDS)**.

addressable storage

All storage space that is user-referenced through an Oracle HSM file system. See **online storage**, **nearline storage**, and **offsite storage**.

admin set

A set of user- and/or group-owned storage that administrators use. Admin sets are typically created to administer storage for projects that involve users from several groups and span multiple files and directories.

archival media

Media that stores copies of the files in an Oracle HSM file system. Archival media can include removable tape cartridges, magneto-optical cartridges, disk file systems configured as archival volumes, and cloud storage volumes.

archival storage

Data storage space created on archival media.

archive set

A collection of files that are copied to archival media together, using a common set of policies and parameters. Set membership determines the number of copies made, the parameters of the copying process, and the media used.

The `archiver.cmd` file defines archive sets by file system, directory location, size, and/or user and group ownership. See the `archiver.cmd` (4) man page for additional details.

archiver

The Oracle HSM program that manages the process of copying files to archival media. See the `archiver` (1m) man page for additional details.

associative staging

Copying a group of files that are no longer resident in the disk cache from archival media back to the Oracle HSM disk cache when a user or application accesses any one member of the group. Associative staging insures that files that are used together are staged together. File owners can associate any files that reside in the same directory by setting the associative-staging attribute on the related files. See the `staging` (1) man page, **staging**, and **disk cache** for additional details.

audit (full)

The process of loading cartridges to verify their volume serial numbers (VSNs). For magneto-optical cartridges, the capacity and space information is determined and entered into the automated library's catalog.

automated library

A device that stores removable media cartridges, loads them into drives, and unloads them without operator intervention. An automated library contains cartridge storage slots, one or more drives, a transport mechanism for the cartridges, and, often, a mechanism for ingesting and exporting cartridges. See **direct-attached automated library**, **network-attached automated library**, **robot**, **transport**.

backup

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

The size of the smallest addressable data unit on a block device, such as a magnetic tape cartridge or hard disk. The block size for a Linear Tape Open (LTO) cartridge is 256 kilobytes. For an Oracle StorageTek T10000 tape cartridge, the block size is 2048 kilobytes. For disk devices, block size is equivalent to *sector size*, which is typically 512 bytes.

buffered I/O

Writing and reading data to storage media, such as magnetic tape or disk, via an intervening segment of host memory, called the *buffer*. When an application writes to the storage device, the host lets the required changes accumulate in memory before writing them out to the media in a single operation, a process called *flushing the buffer*. When an application reads from the media, the host reads more data from the media than the application requested and stores it all in memory, in case the application subsequently requests the additional data.

By consolidating a large number of application I/O requests into a smaller number of hardware I/O operations, buffering improves I/O performance and uses storage hardware more efficiently, even when applications send or request data in suboptimal amounts or at inconsistent rates. Compare **direct I/O**.

cartridge

A container for data-storage media, such as magnetic tape or optical media. Also called a *volume*, a *tape*, a *piece of media*, or, loosely, a *VSN*. See **volume**, **volume serial number (VSN)**.

catalog

The Oracle Hierarchical Storage Manager software's record of the removable media volumes in an automated library. Volumes are identified and tracked using a *volume* *serial number*. See **volume serial number (VSN)**, **historian**.

client-server

Adjective describing a distributed computer application that divides work between *servers* that specialize in providing files or services and *clients* that request files and services when performing particular tasks.

cloud library

Cloud storage that is accessed and managed as if it were a network-attached tape library containing a set of labeled media volumes. For additional information, see the `cloud` (7) man page.

cloud storage

Storage provided as an abstract, network service, without reference to any particular physical implementation or location. Cloud storage supplies users with an agreed level of service rather than a set of defined physical resources. Users and applications store and access data by addressing logical containers rather than physical locations.

A **cloud library** is the Oracle Hierarchical Storage Manager interface to cloud storage.

data device

In a file system, a device or group of devices upon which file data is stored.

Data Integrity Validation (DIV)

Data Integrity Validation, a feature of Oracle StorageTek tape drives that works with the Oracle HSM software to calculate and compare checksums during I/O.

During write operations, Oracle HSM calculates a four-byte checksum for each data block and passes the checksum to the drive along with the data. The tape drive then recalculates the checksum and compares the result to the value supplied by Oracle HSM. If the values agree, the drive writes both the data block and the checksum to tape. Optionally, when the write operation is complete, Oracle HSM can ask the tape drive to rescan the data, recalculate checksums, and compare the results to the checksums stored on the tape.

During read operations, both the drive and Oracle HSM read a data block and its associated checksum from tape. Each recalculates the checksum from the data block and compares the result to the stored checksum. If checksums do not match at any point, the drive notifies Oracle HSM that an error has occurred.

data mover

In an Oracle HSM shared file system, a client that is connected to tape drives and performs tape I/O on behalf of the metadata server. See **distributed I/O**.

DAU

See **disk allocation unit (DAU)**.

device logging

A configurable feature of Oracle HSM that provides specific error information for the hardware devices that support file systems.

device scanner

Software that periodically monitors the presence of all manually mounted removable devices and detects the presence of mounted cartridges that can be requested by users or by other processes.

direct access

Access to files on archival media without preliminary staging to the disk cache. The `-n` (*stage never*) staging attribute marks files for direct access. See **removable media file** and the `stage` (1) man page for more information.

direct-attached automated library

An **automated library** that is connected directly to the host via a SCSI interface. Oracle HSM software can directly control SCSI-attached libraries.

direct I/O

Reading from and writing to a storage device without using memory buffers on the host. Direct I/O can improve performance when transferring large amounts of block-aligned, sequential data. But otherwise, **buffered I/O** generally provides the best results.

directory

A file data structure that points to other files and directories within the file system.

disk allocation unit (DAU)

In QFS file systems, the minimum amount of contiguous space that each I/O operation consumes, regardless of the amount of data written. The disk allocation unit thus determines the minimum number of I/O operations needed when transferring a file of a given size. The DAU should always be a multiple of the block size of the disk device.

The size of the disk allocation unit varies depending upon the QFS device type selected and user requirements. The `md` device type uses dual-allocation units: the DAU is `4` kilobytes for the first eight writes to a file and then a user-specified `16`, `32`, or `64` kilobytes for any subsequent writes, so that small files are written in suitably small blocks, while larger files are written in larger blocks. The `mr` and striped group device types use a DAU that is adjustable in increments of `8` within the range `[8-65528]` kilobytes. Files are thus written in large, uniform blocks that can closely approximate the size of the large, uniformly sized files.

See **block size.**

disk buffer

In Oracle Hierarchical Storage Manager SAM-Remote configurations, the buffer on the SAM-Remote server host that is used for archiving data from the client to the server.

disk cache

The disk-resident portion of an Oracle HSM file system where files are written, modified, and read. New and modified files are copied from the disk cache to archive media and may eventually be released from the disk. When users subsequently request non-resident files, the files are staged (copied) from the archival media to the disk. 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. See **high-water mark**, **low-water mark**, and **releaser**.

disk striping

Writing a file across several disks, thereby improving access performance and increasing overall storage capacity. See also **striping**.

distributed I/O

A feature of Oracle Hierarchical Storage Manager that lets the metadata server of a shared QFS file system delegate tape I/O to file system clients that are connected to tape drives. This reduces loads on the server and makes more efficient use of drives and SAN bandwidth. See **data mover**.

DIV

See **Data Integrity Validation (DIV)**.

drive

1. A electromechanical mechanism for transferring data to and from a removable-media volume, such as a magnetic tape cartridge.
2. An electromechanical, magnetic hard disk drive.
3. A solid-state device that emulates a disk drive. See **solid-state device**.

Ethernet

A packet-switched, local area network technology.

extent array

An array within a file's inode that defines the disk location of each data block assigned to the file.

family set

In Oracle HSM and QFS file-system configurations, a group of physical devices that function as a single logical devices, such as a set of data and metadata disks or an automated library and its associated drives.

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 logical structure that organizes data into a hierarchy of directories and files.

file system directives

In the Oracle HSM `archiver.cmd` file, archiver and releaser directives that are specific to a particular file system. File system directives follow global directives and include all directives between an `fs =` *`filesystem-specifier`* directive and the next `fs =` directive or the end of the file. File system directives override any global directives that may also apply. See the `archiver.cmd` (4) man page for details.

ftp

File Transfer Protocol, a network protocol for transferring files between two hosts. For a more secure alternative, see **sftp**.

global directives

In the Oracle HSM `archiver.cmd` file, archiver and releaser directives that apply to all file systems. Global directives appear before the first `fs =` *`filesystem-specifier`* directive. See the `archiver.cmd` (4) man page for details.

grace period

In a disk **quota**, the amount of time that the file system allows the total size of files belonging to specified user, group, and/or **admin set**s to exceed the **soft limit** specified in the quota.

HA-COTC

High-Availability Clients Outside the Cluster, the failover configuration for the metadata servers of a shared QFS file system that includes clients.

In an HA_COTC configuration, the file system is shared between active and potential QFS metadata servers and file-system clients. The metadata servers are hosted on a two-node, failover cluster. Clients are not hosted on cluster nodes. Solaris Cluster thus software ensures that the metadata servers remain available so that clients can access metadata and obtain I/O licenses. But clients are not configured for failover and cannot therefore compromise the integrity of the file system following a failure.

HA-QFS

High Availability QFS, the failover configuration that insures that a QFS unshared, standalone file system remains accessible in the event of a host failure.

In an HA-QFS configuration, the file system is configured on both nodes of a two-node cluster managed by Solaris Cluster software. At any given time, only one node mounts the QFS file system. If the node that is mounting the file system fails, the clustering software automatically initiates fail over and re-mounts the file system on the remaining node.

HA-SAM

High-Availability Oracle Hierarchical Storage Manager, the failover configuration for an archiving QFS file system.

In an HA-SAM configuration, Oracle Solaris Cluster software maintains the availability of the file system by insuring that the QFS metadata server and the Oracle Hierarchical Storage Manager application continue to operate even if a server host fails. The file system is shared between active and potential QFS metadata servers hosted on a two-node cluster that is managed by Solaris Cluster software and Data Services.

hard limit

In a quota, the amount of time that the file system allows the total size of files belonging to specified user, group, and/or admin set IDs to exceed the soft limit specified in the quota. See **quota**, **admin set**, **soft limit**.

high-water mark

1. The percentage disk-cache utilization at which Oracle HSM starts the releaser process, deleting previously archived files from disk. A properly configured high-water mark insures that the file system always has enough space available for new and newly staged files. For more information, see the `sam-releaser` (1m) and `mount_samfs` (1m) man pages. Compare **low-water mark**.
2. In a removable media library that is part of an archiving file system, the percentage media-cache utilization at which Oracle HSM starts the recycler process. Recycling empties partially full volumes of current data so that they can replaced by new media or relabeled.

historian

The Oracle HSM historian is a catalog of volumes that have been exported from the automated media libraries defined in the `/etc/opt/SUNWsamfs/mcf` file. By default, it is located at `/var/opt/SUNWsamfs/catalog/historian` on the Oracle HSM file-system host. For additional information, see **catalog** and the `historian` (7) man page.

hosts file

The `hosts.`*`filesystem-name`* file that identifies the hosts that can mount a shared QFS file system. See the `hosts.fs` (4) man page for details.

indirect block

A disk block that contains a list of storage blocks. Oracle HSM and QFS file-system metadata can contain 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

An index node, a 512-byte metadata structure that defines a file for the file system. An inode describes all the attributes associated with a file other than the name. The attributes include ownership, access, permissions, size, and the location of the file.

inode file

In a QFS file system, a metadata file called `.inodes` that contains the inode structures for all files resident in the file system.

kernel

The program that provides basic operating 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

In a shared QFS file system, a function that grants a client permission to perform an operation on a file for a specified period of time. The metadata server issues leases to each client. Leases can be renewed as necessary.

library

See **automated library**.

library catalog

See **catalog**.

libsam

An application programing interface (API) library that lets applications manipulate Oracle Hierarchical Storage Manager operations and files stored in StorageTek QFS file systems. With the `libsam` library, software applications that run on the file-system metadata server can access and manipulate file systems using local function calls. See the `intro_libsam` (3) and `rest_libsam` (3) man pages for details. Compare **libsamrpc**, **rest_libsam**.

libsamrpc

An application programing interface (API) library that lets applications manipulate Oracle Hierarchical Storage Manager operations and files stored in StorageTek QFS file systems. The `libsamrpc` library makes remote procedure calls, so calling applications can run on any host on the network. It supports a subset of the `libsam` functions. Compare **libsam**, **rest_libsam**.

Linear Tape File System (LTFS)

An open standard for file systems on magnetic tape media. LTFS provides directory and file metadata that let users and applications use data as if it were stored on magnetic or solid-state disk.

local file system

1. A QFS file system that is not shared with other hosts.
2. A file system that is installed on a server for use by the operating system.
3. A file system that is installed on one node of a Solaris Cluster system and is not made highly available.

low-water mark

In an archiving file system, the percentage disk-cache utilization at which Oracle HSM stops the releaser process and stops deleting previously archived files from disk. A properly configured low-water mark insures that the file system retains as many file in cache as possible, for best performance, while making space available for new and newly staged files. For more information, see the `sam-releaser` (1m) and `mount_samfs` (1m) man pages. Compare **high-water mark**.

LTFS

See **Linear Tape File System (LTFS)**.

LUN

A Logical Unit Number, a logical partition of a physical device that is used as if it were an independent device.

mcf

The Master Configuration File that defines QFS file systems, data and metadata devices, and Oracle HSM archival data devices.

media

Material that stores data. Common storage media include magnetic tape, magnetic disks, solid-state devices, cloud services, and optical disks.

media migration

1. Copying files from one type or generation of archival tape media to a different type or a newer generation of the same type.
2. Copying files from old, worn archival tape media to new, replacement media.
3. A feature of Oracle Hierarchical Storage Manager 6.1 and later that automates the above processes.

metadata

Literally, data about data. In a file system, metadata is information about files and directories. It includes the locations of each file's data on storage media, file attributes such as file type (directory, regular file, character special file, block special file, etc), modification times, ownership, access permissions, and checksums. See **inode**, **indirect block**.

For additional details, see the Oracle HSM `sls` (1) and Solaris `ls` (1) man pages.

metadata device

In an Oracle HSM high-performance (type `ma`) file system, a dedicated storage device type (type `mm`) that stores only file system metadata. See the `mcf` (4) man page.

You can use solid-state disk, electromechanical magnetic disk, or hardware or software mirrored devices as metadata devices.

metadata server (MDS)

The host that controls Oracle Hierarchical Storage Manager and StorageTek QFS file systems. The metadata server manages the file-system metadata, maintains configuration information for file systems and related processes, such as archiving, participates in file-system I/O, and, in shared configurations, makes the file system available to clients.

Only one metadata server can be *active* at a time. But, in shared configurations, you can configure some or all clients as standby, *potential* metadata servers that can be activated should the active server fail or require disruptive maintenance.

mount point

The directory on which a file system is mounted.

multireader file system

An Oracle HSM file system configuration in which all file system hosts can read files but only one host can write them. For more information, see the `mount_samfs` (1m) man page.

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. See **automated library**.

network-attached automated library

A library that is controlled by a software package supplied by the vendor. A parameter file identifies network-attached libraries to the Oracle HSM software, and a special Oracle HSM media changer daemon provides the interface to the vendor software. Oracle StorageTek ACSLS software controls Oracle StorageTek network-attached libraries. See **automated library**.

NFS

Network File System, a file system that provides transparent access to remote file systems on heterogeneous networks.

offsite storage

Storage that is remote from the server and is used for disaster recovery.

online storage

Storage that is immediately available. See **disk cache**.

Oracle HSM

1. A common abbreviation for Oracle Hierarchical Storage Manager.
2. An adjective describing a QFS file system that is configured for archiving and managed by Oracle HSM software.

partition

A portion of a device or a side of a magneto-optical cartridge.

potential metadata server

See **metadata server (MDS)**.

preallocation

The process of reserving a contiguous amount of space on the Oracle HSM 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. See **disk cache**.

pseudo device

A software subsystem or driver with no associated hardware.

QFS

1. A QFS file system, an Oracle UNIX file system that offers high performance and high capacity. QFS file systems can be used on their own or with Oracle Hierarchical Storage Manager.
2. The StorageTek QFS product, which includes the file system without the Oracle Hierarchical Storage Manager software.

qfsdump

See **samfsdump (qfsdump)**.

qfsrestore

See **samfsrestore (qfsrestore)**.

quota

The amount of storage resources that specified user, group, or **admin set**s are allowed to consume. See **hard limit** and **soft limit**.

RAID

Redundant Array of Independent Disks, a disk technology that uses several independent disks to reliably store files. Depending on the architecture, it can protect against data loss should one or more disks fail and can provide higher throughput than individual disks.

recovery point

A compressed file that stores a point-in-time backup copy of the metadata for a Oracle HSM file system.

In the event of a data loss—anything from accidental deletion of a user file to catastrophic loss of a whole file system—an administrator can recover to the last known-good state of the file or file system almost immediately by locating the last recovery point at which the file or file system remained intact. The administrator then restores the metadata recorded at that time and either stages the files indicated in the metadata to the disk cache from archival media or, preferably, lets the file system stage files on demand, as users and applications access them. See **samfsdump (qfsdump)**.

recycler

An Oracle HSM utility that reclaims space when an archival tape volume is largely filled with *stale* copies (copies that no longer reflect the current state of the file). The recycler moves any remaining current file copies to other media and then relabels the volume. The relabeled volume can then be over-written with new files.

recycling

The process of moving current files from an archival media volume and relabeling the media for re-use. For details, see the `sam-recycler` (1m), `recycler.cmd` (1m), and `recycler.sh` (1m) man pages.

regular expression

A string of characters in a standardized pattern-matching language that is designed for searching, selecting, and editing other character strings, such as file names and configuration files. For full details of the regular expression syntax used in Oracle HSM file-system operations, see the Solaris `regex` and `regcmp` man pages.

release priority

The priority according to which the Oracle HSM **releaser** deletes a file from the file **disk cache** once the file has been successfully archived. See the `sam-releaser` (1m) man page for details.

releaser

A Oracle HSM 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. See the `sam-releaser` (1m) and `releaser.cmd` (4) man page for details.

remote procedure call

See **RPC**.

removable media file

In an Oracle HSM file system, 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. See **direct access**.

REST

Representational state transfer, the style of software architecture pioneered by the World Wide Web. REST emphasizes the roles of components, their interactions, and the ways that they represent data, rather than the internal implementation of components. Typical REST applications communicate via Hypertext Transfer Protocol (HTTP). REST is an alternative to RPC. See **RPC**.

rest_libsam

An application programing interface (API) that lets applications control Oracle Hierarchical Storage Manager operations and access files stored in StorageTek QFS file systems. The `rest_libsam` library provides a lightweight **REST** interface that operates over an authenticated HTTPS connection. It is implemented on top of the existing `libsam` library and supports a subset of the `libsam` functions. Compare **libsam**, **libsamrpc**.

robot

An **automated library** component 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, a mechanism that lets a client application execute subroutines that run under an independent server application.

SAM

A common abbreviation for Storage Archive Manager, the former name of the Oracle Hierarchical Storage Manager product.

SAM-Remote

An Oracle HSM client-server configuration that lets an Oracle HSM metadata server access an automated tape library that is controlled by another Oracle HSM metadata server. The client is configured with pseudodevices that represent the devices that the server makes available and uses a specified subset of the archive media on the server.

SAM-QFS

1. A common abbreviation for older versions of the Oracle Hierarchical Storage Manager product.
2. An adjective describing a QFS file system that is configured for archiving and managed by Oracle HSM software.

samfsdump (qfsdump)

An Oracle HSM command that backs up file system metadata to a dump file. See **recovery point**.

If the Oracle Hierarchical Storage Manager packages are not installed, the command is called `qfsdump`.

samfsrestore (qfsrestore)

A program that restores inode and directory information from a recovery point. See **samfsdump (qfsdump)**, **recovery point**.

SAN

Storage Area Network.

SAS

Serial-Attached SCSI.

SC-RAC

Solaris Cluster-Oracle Real Application Cluster (SC-RAC), a high-availability Oracle Database solution that use QFS file systems.

In an SC-RAC solution, Oracle RAC software coordinates I/O requests, distributes workload, and maintains a single, consistent set of database files for multiple Oracle Database instances running on the nodes of a cluster. In the SC-RAC configuration, Oracle Database, Oracle Real Application Cluster (RAC), and QFS software run on two or more of the nodes of a cluster managed by Oracle Solaris Cluster software. One node is configured as the metadata server (MDS) of a QFS shared file system. The remaining nodes are configured as potential metadata servers that share the file system as clients. If the active metadata server node fails, Solaris Cluster software automatically activates a potential metadata server on a healthy node and initiates failover. Since the QFS file system is shared and already mounted on all nodes, access to the data remains uninterrupted.

SCSI

Small Computer System Interface, an electrical communication specification commonly used for peripheral devices such as disk and tape drives and automated libraries.

seeking

Moving the read/write heads of a disk device from one disk location to another during random-access I/O operations.

sftp

Secure File Transfer Protocol, a secure implementation of `ftp`. See **ssh**.

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 -u` command

Small Computer System Interface

See **SCSI**.

soft limit

In a quota, the maximum amount of storage space that a specified user, group, and/or admin set IDs can fill for an indefinite period. Files can use more space than the soft limit allows, up to the hard limit, but only for a short grace period defined in the quota. See **grace period**, **hard limit**, **quota**, **admin set**.

solid-state device

A storage device that uses electronically rewritable, non-volatile, NAND flash memory as the storage medium, such as a SAS-attached, solid-state disk drive (SSD).

Solid-state drives can provide significantly higher inputs and outputs per second (IOPS) and significantly lower latency compared to traditional magnetic hard drives. They are thus particularly good choices for use as the metadata devices of Oracle Hierarchical Storage Manager and StorageTek QFS, high-performance, `ma` file systems.

ssh

Secure Shell, an encrypted network protocol that allows secure, remote command-line login and command execution.

staging

In Oracle HSM file systems, the process of copying an archived file that is no longer resident in the disk cache from archival storage back to the disk cache. See the `staging` (1) and `stager.cmd` (4) man pages, **disk cache**, and **associative staging** for additional information.

Storage Archive Manager

The former name of the Oracle Hierarchical Storage Manager product.

storage slots

In an automated library, the storage bays that hold media cartridges that are not mounted in drives.

stripe size

During striped device access, the number of disk allocation units (DAUs) that a QFS file system writes before moving to the next device in the stripe, as specified by the `stripe=` mount option.

striped group

In a QFS file system, a collection of devices configured as a single logical device of type `gXXX`. See the `mcf` (4) man page for additional information.

striping

Writing files to multiple devices in parallel, so that each file is spread across all the devices.

QFS file systems support two types of striping:

- *Hard striping* is a permanent feature of the file system that you enable when you specify striped group (type `gXXX`) devices in the Master Configuration File (`mcf`) entries that define the file system.
- *Soft striping* is an optional feature that enable or disable when you mount the file system with the `stripe=` mount parameter.

Compare **round-robin**.

SUNW.hasam

A Solaris Cluster Data Services resource type that supports failover for the Oracle Hierarchical Storage Manager application. `SUNW.hasam` is included with the Oracle HSM software. See HA-SAM.

SUNW.HAStoragePlus

A Solaris Cluster Data Services resource type that manages failover of a server's local storage, so that critical state and dynamic configuration information remains available. `SUNW.HAStoragePlus` is included in the Solaris Cluster software as a standard resource type. See HA-QFS, HA-SAM.

SUNW.qfs

A Solaris Cluster Data Services resource type that supports failover for the metadata servers of a high-availability, StorageTek QFS file system. `SUNW.qfs` is included with the Oracle Hierarchical Storage Manager and StorageTek QFS software. See HA-QFS, HA-SAM, and HA-COTC.

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.

transport

See **robot**.

`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

1. On storage media, a single, accessible, logical storage area, usually addressed by a volume serial number (VSN) and/or volume label. Storage disks and magnetic tape cartridges can hold one or more volumes. For use, volumes are *mounted* on a file system at a specified mount point. See **volume serial number (VSN)**, **mount point**.
2. A magnetic tape cartridge that holds a single logical volume. See **cartridge**.
3. On a random-access disk device, a file system, directory or file that is configured and used as if it were a sequential-access, removable-media cartridge, such as a tape.

volume overflow

A capability that enables the system to span a single file over multiple **volume**s. Volume overflow is useful for sites using very large files that exceed the capacity of their individual cartridges.

volume serial number (VSN)

1. A serial number assigned to a tape or disk storage volume. A volume serial number can consist of up to six uppercase, alphanumeric characters, must start with a letter, and must identify the volume uniquely within a given context, such a tape library or partition. The volume serial number is written on the volume label.
2. Loosely, a specific storage volume, especially a removable media cartridge. See **cartridge**, **volume**.

WORM

Write-Once-Read-Many. A storage classification for media that can be written only once but read many times.