

Using Sun QFS and Sun Storage Archive Manager With Oracle® Solaris Cluster

Copyright © 2011, 2012, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée d'The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation.

Contents

Preface	5
1 Using SAM-QFS With Oracle Solaris Cluster	7
About Failover and Highly Available Configurations	7
Task Map: Working With Oracle Solaris Cluster Software	8
2 Requirements for Using SAM-QFS With Oracle Solaris Cluster	11
Basic Product Familiarity	11
Hardware Requirements	11
Software Requirements	12
Disk Device Requirements	12
Requirements for Clustered (Shared) File Systems	12
Requirements for Failover (Local) File Systems	12
Disk Device Redundancy	13
Storage Redundancy	14
Data Path Redundancy	14
Performance Considerations	14
Example - Verifying Devices and Device Redundancy	15
3 Configuring Sun QFS Local Failover File Systems With Oracle Solaris Cluster	19
Task Map: Configuring Local Failover File Systems With Oracle Solaris Cluster	19
Preparing the Host Systems	20
Editing mcf Files for a Local Failover File System	20
▼ How to Prepare to Create a Local Sun QFS File System and Create an mcf File	21
▼ How to Configure a Failover File System as a SUNW.HASStoragePlus Resource	22
▼ How to Verify the Resource Group on All Nodes	23

4	Configuring Sun QFS Shared File Systems With Oracle Solaris Cluster	25
	Task Map: Configuring Clustered File Systems With Oracle Solaris Cluster	25
	Preparing the Host Systems	26
	Editing mcf Files for a Clustered File System	27
	▼ How to Edit mcf Files for a Clustered File System	27
	Creating the Shared Hosts File on the Metadata Server	29
	How Metadata Server Addresses Are Obtained	29
	▼ How to Verify the Daemons	32
	▼ How to Enable a Shared File System as a SUNW.qfs Resource	33
	▼ How to Bring the Shared Resource Online	33
	▼ How to Verify the Resource Group on All Nodes	34
5	Configuring SAM-QFS Archiving in an Oracle Solaris Cluster Environment (HA-SAM)	35
	About HA-SAM	35
	Before You Begin	36
	▼ How to Confirm the Catalog and Stager Symbolic Links	37
	▼ How to Configure SAM-QFS Archiving in an Oracle Solaris Cluster Environment	38
	Using the samd hastop Command	39
	Using StorageTek Libraries with HA-SAM	39
6	Configuring Clients Outside of the Cluster	41
	COTC Task Map	41
	Configuration Requirements	42
	▼ How to Configure the Metadata Server	42
	▼ How to Configure the File System Clients	46
	Configuration Example	49
	Sample Configuration Files for Metadata Server	50
	Client Network Configuration	50
	Troubleshooting an Issue With Oracle RAC	51

Preface

Using Sun QFS and Storage Archive Manager with Oracle Solaris Cluster provides information about using Oracle Solaris Cluster with Sun QFS and Sun Storage Archive Manager.

Who Should Use This Book

This book is intended for anyone who is interested in using Oracle Solaris Cluster with the Sun QFS and Sun Storage Archive Manager software.

Related Books

- *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*
- *Sun QFS File System 5.3 Configuration and Administration Guide*
- *Sun Storage Archive Manager 5.3 Configuration and Administration Guide*
- *Sun QFS and Sun Storage Archive Manager Reference Manual*

Access to Oracle Support

Oracle customers 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.

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P-1 Typographic Conventions

Typeface	Description	Example
AaBbCc123	The names of commands, files, and directories, and onscreen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. <code>machine_name% you have mail.</code>

TABLE P-1 Typographic Conventions (Continued)

Typeface	Description	Example
AaBbCc123	What you type, contrasted with onscreen computer output	machine_name% su Password:
<i>aabbcc123</i>	Placeholder: replace with a real name or value	The command to remove a file is <i>rm filename</i> .
<i>AaBbCc123</i>	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . A <i>cache</i> is a copy that is stored locally. Do <i>not</i> save the file. Note: Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P-2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	\$
Bash shell, Korn shell, and Bourne shell for superuser	#
C shell	machine_name%
C shell for superuser	machine_name#

Using SAM-QFS With Oracle Solaris Cluster

This chapter provides information about using SAM-QFS with Oracle Solaris Cluster. The following Oracle Solaris Cluster versions are supported in the SAM-QFS 5.3 release:

- Oracle Solaris Cluster 3.3 5/11 on Oracle Solaris 10 operating systems
- Oracle Solaris Cluster 4.0 on Oracle Solaris 11 operating systems

About Failover and Highly Available Configurations

The following configuration methods are supported for use in the Oracle Solaris Cluster environment:

- HA-QFS – A Sun QFS highly available local file system.
In this configuration, the Sun QFS file system is mounted locally (not shared) by a single node in the Oracle Solaris Cluster environment. If the node hosting the file system fails, then the Oracle Solaris Cluster software automatically re-mounts the file system on another node within the cluster. The highly available Sun QFS file system uses the `SUNW.HASStoragePlus` resource type, which is distributed with the Oracle Solaris Cluster product. From the Oracle Solaris Cluster perspective, this use is referred to as a failover file system.
- SC-RAC - A shared QFS file system within an Oracle Solaris Cluster environment for use by the Oracle Real Application Cluster (RAC) application.
In this configuration, one of the Oracle Solaris Cluster nodes acts as the metadata server for the file system. If the metadata server fails, the Oracle Solaris Cluster software, in coordination with the RAC application, automatically moves file system operations from the failing metadata server to a different metadata server within the cluster. Because the file system data is already shared and mounted on the failover metadata server, no re-mounting of the file system is required. The shared file system configuration uses the `SUNW.qfs` resource type, which is distributed with the Sun QFS product. From the Oracle Solaris Cluster perspective, this use is known as a clustered file system.
- SC-COTC – A shared QFS file system with failover capability where all clients accessing the data are outside the cluster.

In this configuration, two or more nodes within the Oracle Solaris Cluster handle metadata operations. A single node is the metadata provider, while additional nodes wait in a passive mode as potential metadata servers in case a failover is required.

This is a failover data service configuration, not a scalable data service. For more information, see [Chapter 6, “Configuring Clients Outside of the Cluster.”](#)

- HA-SAM – A high availability shared archiving configuration.

In this configuration, the service is made highly available by the Oracle Solaris Cluster software moving the archiving and metadata service from an active server node to a node waiting in a passive mode. HA-SAM depends on the Sun QFS Oracle Solaris Cluster agent, and therefore this configuration must be installed with a shared file system that is mounted and managed by the Sun QFS Oracle Solaris Cluster agent. This is a failover data service configuration, not a scalable data service. For more information, see [“About HA-SAM” on page 35.](#)

Note – Although installing the file system in an Oracle Solaris Cluster environment improves reliability and decreases or eliminates unplanned downtime, it does not eliminate planned downtime. To maintain the health of the file system, you might still need to shut down the Sun QFS or SAM-QFS software occasionally to run the `samfsck` process. You will also need to shut down the software to apply software patches or updates.

Task Map: Working With Oracle Solaris Cluster Software

Enabling Oracle Solaris Cluster software to work with Sun QFS or SAM-QFS involves the tasks described in the following table.

Step	Task	Description
1	Confirm Oracle Solaris Cluster-specific requirements.	The basic requirements for Sun QFS or SAM-QFS with Oracle Solaris Cluster software are very similar to those for the products without Oracle Solaris Cluster software. However, some specific requirements simplify later steps in the process. For information, see Chapter 2, “Requirements for Using SAM-QFS With Oracle Solaris Cluster.”
2	Install Oracle Solaris Cluster software and configure nodes as appropriate.	If not already done, install and configure Oracle Solaris Cluster software as explained in <i>Oracle Solaris Cluster Software Installation Guide</i> .
3	Verify Sun QFS or SAM-QFS installation requirements.	Additional requirements for installing Sun QFS or SAM-QFS software with Oracle Solaris Cluster software are basically the same as for Sun QFS and SAM-QFS software without Oracle Solaris Cluster software. See Chapter 3, “Preparing for Installation,” in <i>Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide</i>.

Step	Task	Description
4	Install Sun QFS or SAM-QFS software.	See Chapter 6, “Installing and Configuring SAM-QFS Manager,” in <i>Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide</i> or Chapter 5, “Installing Sun QFS and SAM-QFS,” in <i>Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide</i> , as applicable.
5	(Optional) Configure failover on the metadata server for clustered (shared) file systems.	See Chapter 4, “Configuring Sun QFS Shared File Systems With Oracle Solaris Cluster.”
6	(Optional) Configure high availability for archiving file systems (HA-SAM).	See Chapter 5, “Configuring SAM-QFS Archiving in an Oracle Solaris Cluster Environment (HA-SAM).”
7	(Optional) Configure failover for unshared, unarchiving file systems (HA-QFS).	See Chapter 3, “Configuring Sun QFS Local Failover File Systems With Oracle Solaris Cluster.”
8	(Optional) Configure clients that are part of the shared file system but that are not part of the Oracle Solaris Cluster configuration.	See Chapter 6, “Configuring Clients Outside of the Cluster.”

Requirements for Using SAM-QFS With Oracle Solaris Cluster

Consider the requirements described in this chapter before you attempt to configure Oracle Solaris Cluster with Sun QFS or SAM-QFS.

Basic Product Familiarity

You must be familiar with basic Oracle Solaris Cluster concepts and operations.

In an Oracle Solaris Cluster environment, the disk space used by the file system must be configured on storage that is highly available and redundant. Ensure that you have a good understanding of the concepts in the *Oracle Solaris Cluster Concepts Guide*. You should also be familiar with Oracle Solaris Cluster operations. For information on Oracle Solaris Cluster operations, see the following manuals:

- *Oracle Solaris Cluster Software Installation Guide*
- *Oracle Solaris Cluster System Administration Guide*
- *Oracle Solaris Cluster Data Services Planning and Administration Guide*

Hardware Requirements

You must have at least two UltraSPARC or AMD x64 hosts to use as a cluster.

Note – You cannot mix hardware architectures in a shared Oracle Solaris Cluster environment. All of the nodes must be either SPARC or AMD x64.

Software Requirements

The following Oracle Solaris Cluster versions are supported in the SAM-QFS 5.3 release:

- Oracle Solaris Cluster 3.3 5/11 on Oracle Solaris 10 operating systems
- Oracle Solaris Cluster 4.0 on Oracle Solaris 11 operating systems

Each node must have the same Oracle Solaris Cluster software level and Oracle Solaris Cluster patch collection. You must install Sun QFS or SAM-QFS software packages on each node in the cluster that will host a Sun QFS file system.

Disk Device Requirements

The following sections provide disk device requirements for clustered (shared) file systems and for failover (local) file systems.

Requirements for Clustered (Shared) File Systems

To configure a shared file system, use raw device identifier (DID) devices. In the `cldevice show` command output, these appear as `/dev/did/rdisk/dN` devices. The Oracle Solaris Cluster nodes that share the file system must have access to each DID device through a host bus adapter (HBA) direct connection. All devices must be accessible to the file system from all nodes in the Oracle Solaris Cluster environment that mount the shared file system. For more information about DID devices, see the `did(7)` man page.

Note – Requirements for using device identifier are different for clients outside the cluster.

When you specify these devices in your `mc f` file, use the `/dev/did/rdisk/dN` devices from the `cldevice show` command output. For more information about using the `cldevice` command, see “[Example - Verifying Devices and Device Redundancy](#)” on page 15.

Note – The Sun QFS software supports using multi-owner disk sets in Solaris Volume Manager for Oracle Solaris Cluster to obtain redundancy.

Requirements for Failover (Local) File Systems

To configure a local file system for failover, you must use highly available devices. You can use either of the following options:

- Raw devices – Use Oracle Solaris Cluster global devices. Use the output from the `cldevice show` command to determine the names of the global devices and substitute `global` for `did` when specifying the devices in the `mc f` file. Global devices are accessible from all nodes in an Oracle Solaris Cluster environment, even if these devices are not physically attached to all nodes. If all nodes that have a hardware connection to the disk lose their connections, the remaining nodes cannot access the disk. File systems created on global devices are not necessarily highly available.
- Volume manager - Use one of the following:
 - Solaris Volume Manager for Oracle Solaris Cluster for either shared or stand-alone Sun QFS configurations. Such devices are located in `/dev/md`.
 - Veritas Volume Manager (VxVM) for stand-alone QFS configurations only. Such devices are located in `/dev/vx`.

Use the `clsetup` utility to register volume-managed devices with the Oracle Solaris Cluster framework before you configure your file system.

Note – If you use a volume manager, use it only to provide redundancy. For performance reasons, do not use the volume manager to concatenate storage on separate devices. This practice causes the Sun QFS highly available file system to distribute I/O inefficiently across the component devices.

To confirm which devices in your Oracle Solaris Cluster environment are highly available, use the `cldevice show | grep Device` command. This command lists the paths of the devices in the DID configuration file. In the output from the `cldevice show` command, look for devices that have two or more DID devices listed with the identical DID device number. Such devices are highly available in an Oracle Solaris Cluster environment and can also be configured as global devices for a file system, even if they directly connect only to a single node.

I/O requests issued to global devices from a node other than the direct-attached node are issued over the Oracle Solaris Cluster interconnect. These single-node, global devices cease to be available when all nodes that have direct access to the device are unavailable.

Disk Device Redundancy

To implement redundancy, you have the following options:

- If you are configuring a shared file system, you can obtain redundancy through multi-owner disk sets in Solaris Volume Manager for the Oracle Solaris Cluster environment.

- If you are configuring a highly available file system, you can use either the Solaris Volume Manager or the Veritas Volume Manager to obtain mirroring (RAID-1) or striping (RAID-5). For more information about volume sizing and redundancy configurations, see the *Solaris Volume Manager Administration Guide* or your Veritas Volume Manager documentation.

Storage Redundancy

Storage redundancy is achieved by maintenance of extra disk copies of data using mirroring or RAID-1, or parity across several disks using RAID-5 to enable reconstruction of data after a disk failure. When supported by the hardware, these disk configurations enable you to configure the raw devices in an Oracle Solaris Cluster environment without a volume manager. These raw devices are accessible from multiple nodes, so you can issue the `format` command from any node to obtain information on the disks.

Storage redundancy can also be achieved by using software to support mirroring or RAID. This method, however, is not generally suitable for concurrent access from multiple hosts. Oracle Solaris Cluster software supports mirroring of disk volumes (RAID-1 only) through its multi-owner diskset feature with Sun QFS software and Solaris Volume Manager. No other software redundancy is supported.

Data Path Redundancy

Data path redundancy is achieved with multiple HBAs, which are configured from a single node. If your environment includes multiple HBAs for redundancy, be aware that the Sun QFS file systems require multipathing software like the Oracle Solaris I/O multipathing feature (MPxIO) to enable data path redundancy. For more information, see the *Oracle Solaris SAN Configuration and Multipathing Guide*, or see the `stmsboot(1M)` man page.

To determine redundancy, consult the hardware documentation for your disk controllers and disk devices. You need to know whether the disk controller or disk devices that are reported by the `cldevice show` command are on redundant storage. For information, see the storage controller vendor's documentation set and view the current controller configuration.

Performance Considerations

For optimal file system performance, the metadata and file data should be accessible through multiple interconnects and multiple disk controllers. In addition, plan to write file data to separate, redundant, highly available disk devices.

Plan to write your file system's metadata to RAID-1 disks. You can write file data to either RAID-1 or RAID-5 disks.

If you are configuring a Sun QFS highly available local file system and you are using a volume manager, you will get the best performance when the file system is striping data over all controllers and disks, rather than when the volume manager performs the striping. You should use a volume manager only to provide redundancy.

Example - Verifying Devices and Device Redundancy

This example shows how to use output from the `cldevice` command to find the devices in the Oracle Solaris Cluster environment, determine which devices are highly available, and then determine which devices are redundant.

Determining High Availability

The following example shows the use of the `cldevice show` Oracle Solaris Cluster command to list paths of the devices in the DID configuration file for all nodes. In the output from the `cldevice show` command, look for output that shows a device that is visible from two or more nodes and that has the same World Wide Name. These are global devices.

The example uses Oracle's StorageTek T3 arrays in a RAID-5 configuration. The output shows that you can use devices 3 and 4 for configuring the disk cache for a file system.

EXAMPLE 2-1 `cldevice` Command Example

```
ash# cldevice show | grep Device

+=== DID Device Instances ===
DID Device Name:                /dev/did/rdisk/d1
  Full Device Path:              ash:/dev/rdisk/c0t0d0
DID Device Name:                /dev/did/rdisk/d2
  Full Device Path:              ash:/dev/rdisk/c0t6d0
DID Device Name:                /dev/did/rdisk/d3
  Full Device Path:              ash:/dev/rdisk/c6t50020F2300004921d1
  Full Device Path:              elm:/dev/rdisk/c6t50020F2300004921d1
DID Device Name:                /dev/did/rdisk/d4
  Full Device Path:              ash:/dev/rdisk/c6t50020F2300004921d0
  Full Device Path:              elm:/dev/rdisk/c6t50020F2300004921d0
...

_# The preceding output indicates that both ash and elm can
access DID devices {{d3}} and {{d4}}._
_# These disks are highly available._

ash# format /dev/did/rdisk/d4s2
selecting /dev/did/rdisk/d4s2
[disk formatted]

FORMAT MENU:
  disk      - select a disk
  type      - select (define) a disk type
```

EXAMPLE 2-1 cldevice Command Example (Continued)

```

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

<cmd>    - execute <cmd>, then return
quit
format> verify

```

Primary label contents:

```

Volume name = <          >
ascii name  = <SUN-T300-0118 cyl 34530 alt 2 hd 192 sec 64>
pcyl       = 34532
ncyl       = 34530
acyl       = 2
nhead      = 192
nsect      = 64

```

Part	Tag	Flag	Cylinders	Size	Blocks
0	usr	wm	0 - 17264	101.16GB	(17265/0/0) 212152320
1	usr	wm	17265 - 34529	101.16GB	(17265/0/0) 212152320
2	backup	wu	0 - 34529	202.32GB	(34530/0/0) 424304640
3	unassigned	wu	0	0	(0/0/0) 0
4	unassigned	wu	0	0	(0/0/0) 0
5	unassigned	wu	0	0	(0/0/0) 0
6	unassigned	wu	0	0	(0/0/0) 0
7	unassigned	wu	0	0	(0/0/0) 0

Analyzing the Output From the Commands

The `cldevice show` command in this example lists device `/dev/rdisk/c6t50020F2300004921d0`, which is DID device `/dev/did/rdisk/d4` or global device `/dev/global/rdisk/d4`. This device has two partitions (0 and 1), each of which yields 212152320 blocks for use by a Sun QFS highly available local file system as `/dev/global/rdisk/d4s0` and `/dev/global/rdisk/d4s1`.

You need to issue the `cldevice show` and `format` commands for all devices to be configured for use by the Sun QFS highly available local file system.

- If you want to configure a Sun QFS shared file system on a cluster, you must use highly available, redundant devices.
- If you want to configure a Sun QFS highly available local file system and the `cldevice show` command output indicates that the devices you want to use are JBOD (just a bunch of disks) or dual-port SCSI disk devices, you need to use a volume manager that is supported in an

Oracle Solaris Cluster environment to obtain redundancy. The options available and capabilities provided by such a volume manager are beyond the scope of this manual.

For more information about configuring devices that are on redundant storage, see the Oracle Solaris Cluster software installation documentation.

Configuring Sun QFS Local Failover File Systems With Oracle Solaris Cluster

This chapter contains instructions for configuring the Sun QFS software in an Oracle Solaris Cluster environment. These instructions are for HA-QFS configuration only. Before carrying out the configuration procedures in this chapter, you must have installed the Sun QFS software as described in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*.

The Sun QFS file system can work with Oracle Solaris Cluster in one of the following ways:

- As a highly available shared file system, simply referred to as a *clustered* file system. For information about configuring a clustered file system, see [Chapter 4, “Configuring Sun QFS Shared File Systems With Oracle Solaris Cluster.”](#)
- As a highly available local file system, also known as a *failover* file system.

Task Map: Configuring Local Failover File Systems With Oracle Solaris Cluster

Use the tasks shown in the following table to configure local failover file systems with Oracle Solaris Cluster.

Step	Task	Description
1	Prepare the host systems.	Verify user and group IDs, and NFS-share the file system as described in “Preparing the Host Systems” on page 20.
2	Edit the mcf file.	Edit the mcf file for each host that you want to include in the Oracle Solaris Cluster environment. See “Editing mcf Files for a Local Failover File System” on page 20.
3	Make the highly available file system available to Oracle Solaris Cluster.	Configure the resource as described in “How to Configure a Failover File System as a SUNW.HAStoragePlus Resource” on page 22.

Step	Task	Description
4	Verify the resource group on all nodes.	From any node in the cluster, move the highly available resource to another node as described in “How to Verify the Resource Group on All Nodes” on page 23.

Preparing the Host Systems

To prepare the host systems for a shared file system in an Oracle Solaris Cluster environment, perform the following tasks:

1. Verify that all the hosts have the same user and group IDs.
 - If you are not running the Network Information Name service (NIS), make sure that all `/etc/passwd` and all `/etc/group` files are identical.
 - If you are running NIS, the `/etc/passwd` and `/etc/group` files should already be identical.

For more information, see the `nis+(1)` man page.

2. NFS-share the file system.

The following is a general description of how to NFS-share a file system in an Oracle Solaris Cluster environment. For more information on NFS-sharing file systems that are controlled by `HASStoragePlus`, see *Sun Cluster Data Service for Network File System (NFS) Guide*.

- a. Locate the `dfstab.resource-name` file.

The `Pathprefix` property of `HASStoragePlus` specifies the directory in which the `dfstab.resource-name` file resides.

- b. Add a `share` command to the `Pathprefix/SUNW.nfs/dfstab.resource-name` file.

For example:

```
share -F nfs -o rw /global/qfs1
```

Editing mcf Files for a Local Failover File System

The lines that define a particular file system must be identical in the `mcf` files on all host systems that support the file system. Only one `mcf` file can reside on a host. Because you can have other, additional file systems defined in an `mcf` file, the `mcf` files on different hosts might not be identical.

Note – If you update a metadata server's mcf file at any time after the shared file system is mounted, you must also update the mcf files as necessary on all hosts that can access that shared file system.

▼ How to Prepare to Create a Local Sun QFS File System and Create an mcf File

- 1 Log in to the Oracle Solaris Cluster node.
- 2 Become superuser.
- 3 Use the `format` utility to lay out the partitions on `/dev/global/dsk/d4`.

```
# format /dev/global/rdisk/d4s2
```

```
# format> partition
```

```
[ output deleted ]
```

```
# partition> print
```

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

```
Total disk cylinders available: 34530 + 2 (reserved cylinders)
```

Part	Tag	Flag	Cylinders	Size	Blocks
0	unassigned	wm	1 - 3543	20.76GB	(3543/0/0) 43536384
1	unassigned	wm	3544 - 34529	181.56GB	(30986/0/0) 380755968
2	backup	wu	0 - 34529	202.32GB	(34530/0/0) 424304640
3	unassigned	wu	0	0	(0/0/0) 0
4	unassigned	wu	0	0	(0/0/0) 0
5	unassigned	wu	0	0	(0/0/0) 0
6	unassigned	wu	0	0	(0/0/0) 0
7	unassigned	wu	0	0	(0/0/0) 0

NOTE: Partition 2 (backup) will not be used and was created by `format(1m)` by default.

Partition (or slice) 0 skips over the volume's Volume Table of Contents (VTOC) and is then configured as a 20-gigabyte partition. The remaining space is configured into partition 1.

4 Replicate the global device d4 partitioning to global devices d5 through d7.

This example shows the command for global device d5:

```
# prtvtoc /dev/global/rdisk/d4s2 | fmthard \
-s - /dev/global/rdisk/d5s2
```

5 On all nodes that are potential hosts of the file system, perform the following:

a. Configure the eight partitions (four global devices, with two partitions each) into a Sun QFS file system by adding a new file system entry to the mcf file.

```
# cat >> /etc/opt/SUNWsamfs/mcf <<EOF
```

```
#
# StorageTek QFS file system configurations
#
# Equipment          Equipment      Equipment      Family          Device  Additional
# Identifier         Ordinal       Type           Set             State   Parameters
# -----
qfsnfs1              100           ma             qfsnfs1        on
/dev/global/dsk/d4s0 101           mm             qfsnfs1
/dev/global/dsk/d5s0 102           mm             qfsnfs1
/dev/global/dsk/d6s0 103           mm             qfsnfs1
/dev/global/dsk/d7s0 104           mm             qfsnfs1
/dev/global/dsk/d4s1 105           mr             qfsnfs1
/dev/global/dsk/d5s1 106           mr             qfsnfs1
/dev/global/dsk/d6s1 107           mr             qfsnfs1
/dev/global/dsk/d7s1 108           mr             qfsnfs1
EOF
```

b. Validate that the configuration information you added to the mcf file is correct, and fix any errors in the mcf file before proceeding.

It is important to complete this step before you configure the Sun QFS file system under the HASToragePlus resource type.

```
# /opt/SUNWsamfs/sbin/sam-fsd
```

▼ How to Configure a Failover File System as a SUNW.HASToragePlus Resource

Perform this task if you are configuring a local file system to failover on an Oracle Solaris Cluster platform.

- **Use the `scrgadm` command to set the `FilesystemCheck` command property of `HASStoragePlus` to `/bin/true`.**

All other resource properties for `HASStoragePlus` apply as specified in `SUNW.HASStoragePlus`.

The following example command shows how to use the `scrgadm` command to configure a `SUNW.HASStoragePlus` resource:

```
# scrgadm -a -g qfs-rg -j ha-qfs -t SUNW.HASStoragePlus \
-x FilesystemMountPoints=/global/qfs1 \
-x FilesystemCheckCommand=/bin/true
```

▼ How to Verify the Resource Group on All Nodes

This task ensures that the file system can move from node to node when the Oracle Solaris Cluster software performs a failover.

Perform these steps for each node in the cluster, with a final return to the original server.

- 1 **From any node in the Oracle Solaris Cluster environment, use the `scswitch` command to move the file system resource from one node to another.**

For example:

```
server# scswitch -z -g qfs-rg -h elm
```

- 2 **Use the `scstat` command to verify that the file system resource was moved successfully.**

For example:

```
server# scstat
-- Resources --
Resource Name      Node Name  State      Status Message
-----
Resource: qfs-res  ash       Offline   Offline
Resource: qfs-res  elm       Online    Online
Resource: qfs-res  oak       Offline   Offline
```


Configuring Sun QFS Shared File Systems With Oracle Solaris Cluster

This chapter contains instructions for configuring the shared Sun QFS software in an Oracle Solaris Cluster environment. These instructions are for the following configurations:

- Oracle Real Application Cluster (SC-RAC) – the only scalable application that the Sun QFS software supports in Oracle Solaris Cluster environments. For more information, see *Oracle Solaris Cluster Data Service for Oracle Real Application Clusters*. Also see, “[Troubleshooting an Issue With Oracle RAC](#)” on page 51.
- SAM-QFS archiving (HA-SAM) – For more information, see Chapter 5, Configuring SAM-QFS Archiving in an Oracle Solaris Cluster Environment (HA-SAM) [Chapter 5, “Configuring SAM-QFS Archiving in an Oracle Solaris Cluster Environment \(HA-SAM\)”](#).
- Shared QFS with clients outside the cluster (SC-COTC) – all clients exist outside of the Oracle Solaris Cluster to access data. The metadata operations are handled by a node inside the cluster with an optional passive metadata node available for failover capabilities.

Before carrying out the configuration procedures in this chapter, you must have installed the Sun QFS software as described in [Chapter 5, “Installing Sun QFS and SAM-QFS,”](#) in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide*.

You can also add a shared file system to an existing Oracle Solaris Cluster environment.

Task Map: Configuring Clustered File Systems With Oracle Solaris Cluster

Use the tasks in the table to configure clustered file systems with Oracle Solaris Cluster.

Step	Task	Description
1	Prepare the host systems.	Verify user and group IDs, and NFS-share the file system as described in “Preparing the Host Systems” on page 26.

Step	Task	Description
2	Edit the <code>mc f</code> file.	Edit the <code>mc f</code> file for each host that you want to include in a shared file system in an Oracle Solaris Cluster environment. See “Editing <code>mc f</code> Files for a Clustered File System” on page 27.
3	Create the shared hosts file.	On the metadata server, create the shared hosts file as described in “Creating the Shared Hosts File on the Metadata Server” on page 29.
4	Verify the resource group on all nodes.	On each host that can mount the file system, verify that the file system is mounted and shared. See “How to Verify the Daemons” on page 32.
5	Make the shared file system available to Oracle Solaris Cluster.	Log in to the metadata server and configure the resource as described in “How to Enable a Shared File System as a <code>SUNW . qfs</code> Resource” on page 33.
6	Bring the shared resource online.	On the metadata server, move the shared resource to another node as described in “How to Bring the Shared Resource Online” on page 33.
7	Verify the resource group on all nodes.	On each node in the cluster that is part of the shared file system, move the shared resource to another node as described in “How to Verify the Resource Group on All Nodes” on page 23. When you are finished, return to the metadata server.

Preparing the Host Systems

To prepare the host systems for a shared file system in an Oracle Solaris Cluster environment, perform the following tasks:

- Verify that all the hosts have the same user and group IDs.
 - If you are not running the Network Information Name service (NIS), make sure that all `/etc/passwd` and all `/etc/group` files are identical.
 - If you are running NIS, the `/etc/passwd` and `/etc/group` files should already be identical.

For more information, see the `nis+(1)` man page.

- NFS-share the file system.

This step provides a general description of how to NFS-share a file system in an Oracle Solaris Cluster environment. For more information on NFS-sharing file systems that are controlled by `HASStoragePlus`, see *Oracle Solaris Cluster Data Service for Network File System (NFS) Guide* and the NFS documentation.

- Locate the `dfstab . resource-name` file.

The `Pathprefix` property of `HASStoragePlus` specifies the directory in which the `dfstab . resource-name` file resides.

- b. Add a share command to the *Pathprefix/SUNW.nfs/dfstab.resource-name* file.

For example:

```
share -F nfs -o rw /global/qfs1
```

Editing mcf Files for a Clustered File System

The lines that define a particular file system must be identical in the mcf files on all host systems that support the file system. Only one mcf file can reside on a host. Because you can have other, additional file systems defined in an mcf file, the mcf files on different hosts might not be identical.

Note – If you update a metadata server's mcf file at any time after the shared file system is mounted, you must also update the mcf files as necessary on all hosts that can access that shared file system.

▼ How to Edit mcf Files for a Clustered File System

Perform this procedure for each host that you want to include in a shared file system in an Oracle Solaris Cluster environment.

- 1 **Log in to the host.**
- 2 **Become superuser.**
- 3 **Ensure that an mcf file on this node contains lines that define the new file system copied from the primary node's mcf file.**
 - If an mcf file does not exist on this node, create one and include the file system lines.
 - If an mcf file already exists on this node, add the lines for the new file system.
- 4 **Copy the lines that define the file system from the primary node's mcf file to this node's mcf file.**
- 5 **Use the `samfsconfig` command to locate the local device names for each additional host to be configured in the shared file system.**

The `samfsconfig` command generates configuration information that can help you to identify the devices included in the shared file system. Issue a separate `samfsconfig` command on each client host. Note that the controller number might not be the same controller number as on the metadata server because the controller numbers are assigned by each client host.

6 Update the mcf file on other client hosts.

Use `vi` or another editor to edit the `mcf` file on one of the client host systems. The `mcf` file must be updated on all client hosts to be included in the shared file system. The file system and disk declaration information must have the same data for the Family Set name, Equipment Number, and Equipment Type fields as the configuration on the metadata server. The `mcf` files on the client hosts must also include the shared keyword. The device names, however, can change as controller assignments can change from host to host.

To access or mount a shared file system, a host system must have that file system defined in its `mcf` file. The content of the `mcf` files varies depending on the type of Oracle Solaris Cluster hosts:

- The primary metadata server
- Potential metadata servers

All hosts can be metadata servers because the Oracle Solaris Cluster software fails over system resources in the event of a node failure.

Note – Clients within the cluster are supported within the SC-RAC configuration only. Other configurations have clients within the cluster that serve as passive metadata servers only. They exist as potential metadata servers for failover capabilities but are not supported for reading or writing data as a shared Sun QFS client.

Example 4-1 mcf File Using Hardware RAID and Oracle Solaris Cluster

The following sample `mcf` file includes six shared file systems.

```
psfb1% more /etc/opt/SUNWsamfs/mcf
OraHomes 20 ma OraHomes - shared
/dev/did/dsk/d1s0 21 mm OraHomes -
/dev/did/dsk/d2s0 22 mr OraHomes -
/dev/did/dsk/d3s0 23 mr OraHomes -

CrsHomes 30 ma CrsHomes - shared
/dev/did/dsk/d6s0 31 mm CrsHomes -
/dev/did/dsk/d7s0 32 mr CrsHomes -

OraData 40 ma OraData - shared
/dev/did/dsk/d8s0 41 mm OraData -
/dev/did/dsk/d9s0 42 mr OraData -
/dev/did/dsk/d10s0 43 mr OraData -
/dev/did/dsk/d11s0 44 mr OraData -

votingdisk1 50 ms votingdisk1 - shared
/dev/did/dsk/d4s1 51 md votingdisk1 -
votingdisk2 60 ms votingdisk2 - shared
/dev/did/dsk/d4s2 61 md votingdisk2 -
votingdisk3 70 ms votingdisk3 - shared
/dev/did/dsk/d4s3 71 md votingdisk3 -
```

Creating the Shared Hosts File on the Metadata Server

The shared hosts configuration file must reside in the following location on the metadata server:

```
/etc/opt/SUNWsamfs/hosts.fsname
```

Comments are permitted in the hosts configuration file. Comment lines must begin with a pound character (#). Characters to the right of the pound character are ignored.

The following table shows the fields in the hosts configuration file.

Field	Content
Host Name	This field must contain the alphanumeric name of a metadata server or potential metadata server that is part of the Sun QFS shared file system.
Host Interfaces	<p>This field must contain a comma-separated list of host interface addresses. This field can be created from the output received from the <code>ifconfig -a</code> command. The individual interfaces can be specified in one of the following ways:</p> <ul style="list-style-type: none"> ■ Dotted-decimal IP address form ■ IP version 6 hexadecimal address form ■ As a symbolic name that the local domain name service (DNS) can resolve to a particular host interface <p>Each host uses this field to determine whether it will try to connect to the specified host interface. The system evaluates the addresses from left to right, and the connection is made using the first responding address in the list that is also included in the shared hosts file.</p>

In a shared file system, each client host obtains the list of metadata server IP addresses from the metadata server host.

The metadata server and the client hosts use both the `/etc/opt/SUNWsamfs/hosts.fsname` file on the metadata server and the `hosts.fsname.local` file on each client host (if it exists) to determine the host interface to use when accessing the file system.

How Metadata Server Addresses Are Obtained

The information in this section might be useful when you are debugging.

In a shared file system, each client host obtains the list of metadata server IP addresses from the shared hosts file.

The metadata server and the client hosts use the shared hosts file on the metadata server and the `hosts.fsname.local` file on each client host (if it exists) to determine the host interface to use when accessing the metadata server.

Note – The term client, as in network client, is used to refer to both client hosts and the metadata server host.

This process is as follows:

1. The client obtains the list of metadata server host IP interfaces from the file system's on-disk shared hosts file. To examine this file, issue the `smsharefs` command from the metadata server or from a potential metadata server.
2. The client searches for an `/etc/opt/SUNWsamfs/hosts.fsname.local` file. Depending on the outcome of the search, one of the following activities occur:
 - If a `hosts.fsname.local` file does not exist, the client attempts to connect, in turn, to each address in the servers line in the shared hosts file until it succeeds in connecting.
 - If the `hosts.fsname.local` file exists, the client performs the following tasks:
 - a. It compares the list of addresses for the metadata server from both the shared hosts file on the file system and the `hosts.fsname.local` file.
 - b. It builds a list of addresses that are present in both places, and then it attempts to connect to each of these addresses, in turn, until it succeeds in connecting to the server. If the order of the addresses differs in these files, the client uses the ordering in the `hosts.fsname.local` file.

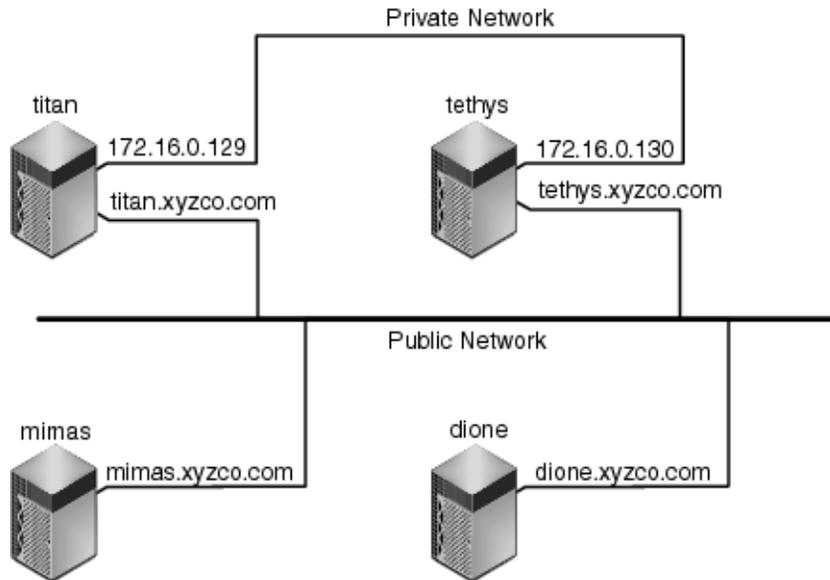
EXAMPLE 4-2 Sun QFS Shared File System Hosts File Example

This set of examples shows a detailed scenario for a shared file system that comprises four hosts.

The following example shows a hosts file that lists four hosts.

```
# File /etc/opt/SUNWsamfs/hosts.sharefs1
# Host Host IP Server Not Server
# Name Addresses Priority Used Host
# ----
titan 172.16.0.129 1 - server
tethys 172.16.0.130 2 -
mimas mimas - -
dione dione - -
```

FIGURE 4-1 Network Interfaces for Sun QFS Shared File System Hosts File Example



EXAMPLE 4-3 File hosts.sharefs1.local on titan and tethys

Systems titan and tethys share a private network connection with interfaces 172.16.0.129 and 172.16.0.130. To guarantee that titan and tethys always communicate over their private network connection, the system administrator has created identical copies of `/etc/opt/SUNWsamfs/hosts.sharefs1.local` on each system.

The following example shows the information in the `hosts.sharefs1.local` files on titan and tethys.

```
# This is file /etc/opt/SUNWsamfs/hosts.sharefs1.local
# Host Name Host Interfaces
# -----
titan 172.16.0.129
tethys 172.16.0.130
```

EXAMPLE 4-4 File hosts.sharefs1.local on mimas and dione

Systems mimas and dione are not on the private network. To guarantee that they always connect to titan and tethys through titan's and tethys's public interfaces, the system administrator has created identical copies of `/etc/opt/SUNWsamfs/hosts.sharefs1.local` on mimas and dione.

The following example shows the information in the `hosts.sharefs1.local` files on mimas and dione.

EXAMPLE 4-4 File hosts.sharefs1.local on mimas and dione (Continued)

```
# This is file /etc/opt/SUNWsamfs/hosts.sharefs1.local
# Host Name Host Interfaces
# -----
titan titan
tethys tethys
```

▼ How to Verify the Daemons

Perform this task on each host that can mount the file system.

1 Verify that the file system is mounted.

If it is not mounted, see “Mounting the File System” in *Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide* and follow the instructions there.

2 Use the ps and grep commands to determine whether the sam-sharefsd daemon is running for this file system.

For example:

```
# ps -ef | grep sam-sharefsd
root 26167 26158 0 18:35:20 ? 0:00 sam-sharefsd sharefs1
root 27808 27018 0 10:48:46 pts/21 0:00 grep sam-sharefsd
```

This example shows that the sam-sharefsd daemon is active for the sharefs1 file system.

Note – If the sam-sharefsd daemon is active for your shared file system, you need to perform some diagnostic procedures.

3 If the output from this command indicates that the sam-sharefsd daemon is not running, determine whether the sam-fsd daemon is running as follows:

a. Use the ps and grep commands to verify that the sam-fsd daemon is running for this file system.

b. Examine the output.

The following code example shows sam-fsd output that indicates that the daemon is running.

```
cur% ps -ef | grep sam-fsd
user1 16435 16314 0 16:52:36 pts/13 0:00 grep sam-fsd
root 679 1 0 Aug 24 ? 0:00 /usr/lib/fs/samfs/sam-fsd
```

c. Do one of the following:

- If the output indicates that the sam-fsd daemon is not running, and if no file system has been accessed since the system's last boot, issue the samd config command, as follows:

- ```
samd config
```
- If the output indicates that the `sam-fsd` daemon is running, enable tracing in the `defaults.conf` file and check the following files to determine whether configuration errors are causing the problem:
    - `/var/opt/SUNWsamfs/trace/sam-fsd`
    - `/var/opt/SUNWsamfs/trace/sam-sharefsd`

## ▼ How to Enable a Shared File System as a SUNW.qfs Resource

- 1 Log in to the metadata server as superuser.
- 2 Use the `scrgadm -p` command to search for the `SUNW.qfs` resource type. For example:
 

```
metadataserver# scrgadm -p | grep SUNW.qfs
```

- 3 If the `SUNW.qfs` resource type is missing, issue the following command:

```
metadataserver# scrgadm -a -t SUNW.qfs
```

- 4 Use the `scrgadm` command to register and configure the `SUNW.qfs` resource type.

The `SUNW.qfs` resource type is part of the Sun QFS software package. Configuring the resource type for use with your shared file system makes the shared file system's metadata server highly available. Oracle Solaris Cluster scalable applications can then access data contained in the file system.

The following code example shows how to use the `scrgadm` command to register and configure the `SUNW.qfs` resource type. In this example, the nodes are `scnode-A` and `scnode-B`. `/global/sharefs1` is the mount point as specified in the `/etc/vfstab` file.

```
scrgadm -a -g qfs-rg -h scnode-A,scnode-B
scrgadm -a -g qfs-rg -t SUNW.qfs -j qfs-res \
 -x QFSFileSystem=/global/sharefs1
```

---

**Note** – In a SAM-QFS environment, you can also configure the archiving features for high availability using Oracle Solaris Cluster software. For instructions, see [Chapter 5, “Configuring SAM-QFS Archiving in an Oracle Solaris Cluster Environment \(HA-SAM\)”](#).

---

## ▼ How to Bring the Shared Resource Online

- 1 Verify that the file system is mounted on all nodes.

---

**Note** – If you are using the SUNW.qfs resource type, you cannot use the `bg mount` option in the `/etc/vfstab` file.

---

- 2 **Log in to the node upon which the file system is based.**
- 3 **Use the `scswitch` command to bring the Sun QFS resource group online.**

For example:

```
metadataserver# scswitch -Z -g qfs-rg
```

- 4 **Use the `scstat` command to verify that the Sun QFS resource group is online.**

For example:

```
metadataserver# scstat
```

<information deleted from this output>

```
-- Resources --
Resource Name Node Name State Status Message

Resource: qfs-res ash Online Online
Resource: qfs-res elm Offline Offline
Resource: qfs-res oak Offline Offline
```

## ▼ How to Verify the Resource Group on All Nodes

Perform these steps for each node in the cluster, with a final return to the metadata server.

- 1 **From any node in the Oracle Solaris Cluster environment, use the `scswitch` command to move the file system resource from one node to another.**

For example:

```
server# scswitch -z -g qfs-rg -h elm
```

- 2 **Use the `scstat` command to verify that the file system resource was moved successfully.**

For example:

```
server# scstat
-- Resources --
Resource Name Node Name State Status Message

Resource: qfs-res ash Offline Offline
Resource: qfs-res elm Online Online
Resource: qfs-res oak Offline Offline
```

# Configuring SAM-QFS Archiving in an Oracle Solaris Cluster Environment (HA-SAM)

---

This chapter provides information about configuring SAM-QFS Archiving in an Oracle Solaris cluster Environment (HA-SAM)

## About HA-SAM

High availability Sun Storage Archive Manager (HA-SAM) is an interface between a Sun QFS file system and Oracle Solaris Cluster software running on Oracle Solaris for SPARC and x64 hardware. The HA-SAM Oracle Solaris Cluster agent periodically monitors the health of SAM-QFS archiving operations on the primary node. In the event of an unrecoverable problem, the agent switches the SAM-QFS archiving and staging operations to a healthy node. Both voluntary and involuntary failover are supported on active-passive configurations. Only two-node active-passive configurations are supported.

For tape archiving and staging to continue after failover, tape drives must be visible to all nodes in a cluster on which HA-SAM is running, but they should not be configured as SAM-QFS shared drives. HA-SAM also supports disk archiving with disk archives visible to all nodes in a cluster.

HA-SAM depends on the Sun QFS Oracle Solaris Cluster agent and assumes that the Sun QFS file systems are mounted and managed by the Sun QFS agent. HA-SAM requires that the SAM-QFS catalog and stager directories be linked from the standard location to a directory in an HAStoragePlus file system. HA-SAM should be a resource in a resource group that contains Sun QFS and catalog resources.

In order to configure the HA-SAM Oracle Solaris Cluster agent, the Sun QFS Oracle Solaris Cluster agent must already be configured. These instructions assume that you have Sun QFS configured in an Oracle Solaris Cluster environment using a shared Sun QFS file system as described in [Chapter 4, “Configuring Sun QFS Shared File Systems With Oracle Solaris Cluster.”](#)

## Before You Begin

Be aware of the following requirements and restrictions before configuring this feature:

- The nodes on which HA-SAM will be configured must be running the same version of Oracle Solaris OS and Oracle Solaris Cluster. Mixed versions of Oracle Solaris OS or Oracle Solaris Cluster software are not supported. No operating system other than Oracle Solaris is supported.
- The nodes in the cluster running HA-SAM must have the same type of architecture: SPARC or x64. Mixed architectures are not supported.
- No more than two nodes can be configured.
- Only active-passive configurations are supported, not active-active.
- Active I/O to HA-SAM file systems is supported only on the active node of an HA-SAM file system.
- Only shared QFS file systems are supported. Both `ma` and `ms` type file systems are supported. Stand-alone QFS environments are not supported.
- No software volume managers are supported with this configuration.
- Within the HA-SAM environment, the `nosam` mount option must be specified for any non-HA-SAM QFS file systems.
- The HA-SAM resource, QFS file systems, and `HAStoragePlus` file systems must all be configured within the same resource group. A separate Oracle Solaris Cluster resource group must be created for non-HA-SAM file systems.
- When using the `SUNW.hasam` resource type, you cannot specify the `bg` mount option in the `/etc/vfstab` file.
- Fibre tape drives are required. Tape drives must be visible to all systems through the fibre fabric, but should not be configured as SAM-QFS shared drives.
- Disk volumes for disk archiving must be visible to all nodes.
- The active metadata server and potential metadata server must not be configured as a SAM-Remote client or server.
- The SAM-QFS catalog and stager directory must be in the default location: `/var/opt/SUNWsamfs/`. If it is any other location, the cluster nodes will not be able to locate it.
- Before configuring HA-SAM, verify that all SAM-QFS archiving operations are working correctly on the required nodes in the cluster.
- Only highly available (HA) agents are supported; no scalable agents are supported.
- Oracle database software is not supported with this configuration, but the HA-SAM agent is supported as an RMAN target for Oracle backups.
- Voluntary failover forces a failover after 5 minutes in order to avoid problems with other potential HA agents.

- During voluntary failover, active archiving and staging times-out and terminates after less than 5 minutes.
- During failover, all SAM-QFS file systems must be failed over in order to avoid complications with the recycler that could result in loss of data.
- StorageTek network attached tape libraries are supported but not ADIC or Sony 8400 PetaSite Series automated tape libraries.
- HA-SAM environments cannot be managed by the SAM-QFS Manager browser interface.
- During involuntary failover, error messages for idled tape drives may be seen in the `/var/adm/messages` log file. These messages can safely be ignored. They simply indicate that HA-SAM did not have time to idle the drives before failover.
- After failover, volumes in the catalog may be marked with the E flag. This can occur when the software is unable to write the correct label at the end of a tape. To clear this error, see [“How to Clear Media Errors” in Sun Storage Archive Manager 5.3 Configuration and Administration Guide](#).

## ▼ How to Confirm the Catalog and Stager Symbolic Links

In order to configure for high availability, the SAM-QFS catalog and stager queues must be in an HASStoragePlus file system. During the software installation process symbolic links should have been created from `/var/opt/SUNWsamfs` to the shared HASStoragePlus file system. This must be confirmed before continuing with the configuration process.

- 1 **Verify that the `/var/opt/SUNWsamfs/catalog` and `/var/opt/SUNWsamfs/stager` contain a symbolic link to the HASStoragePlus mount point.**

- 2 **View the catalog and stager files.**

```
ls -l /var/opt/SUNWsamfs/catalog /var/opt/SUNWsamfs/stager
```

- 3 **Verify that the output shows symbolic links similar to the following example.**

```
lrwxrwxrwx 1 root other 19 Sep 30 11:05
/var/opt/SUNWsamfs/catalog -> /sam_shared/catalog
lrwxrwxrwx 1 root other 18 Sep 30 11:05
/var/opt/SUNWsamfs/stager -> /sam_shared/stager
```

- **If the symbolic links exist, proceed to [“How to Configure SAM-QFS Archiving in an Oracle Solaris Cluster Environment” on page 38](#).**
  - **If the links do not exist, proceed to the next step.**
- 4 **Copy any existing SAM-QFS catalog information to a temporary location.**

- 5 Delete the existing catalog and stager files from their current location.

```
rm -rf /var/opt/SUNWsamfs/catalog /var/opt/SUNWsamfs/stager
```

- 6 Create the `/var/opt/SUNWsamfs/catalog` and `/var/opt/SUNWsamfs/stager` symbolic links to the `HASStoragePlus` mount point.

The following example is for an `HASStoragePlus` file system mounted at `/sam_shared`.

```
ln -s /sam_shared/catalog /var/opt/SUNWsamfs/catalog
ln -s /sam_shared/stager /var/opt/SUNWsamfs/stager
```

- 7 Copy the catalog information saved in Step 3 to the new shared catalog.

```
cp -rp /var/tmp/catalog/* /var/opt/SUNWsamfs/catalog
```

## ▼ How to Configure SAM-QFS Archiving in an Oracle Solaris Cluster Environment

In this example procedure, two file systems are configured: `qfs1` and `qfs2`.

- 1 Create an HA-SAM resource group.

```
clresourcegroup create -n scnode-A,scnode-B SAM-HA
```

- 2 Create and configure the `SUNW.qfs` resource.

```
clresource create -g SAM-HA -t SUNW.qfs
-x QFSFileSystem=/global/qfs1,/global/qfs2 qfs-res
```

- 3 Create and configure a SAM-QFS shared resource.

```
clresource create -g SAM-HA -t SUNW.HASStoragePlus
-x FilesystemMountPoints=/sam_shared -x AffinityOn=TRUE sam-hastp
```

- 4 Create the HA-SAM resource type.

```
clresourcetype register SUNW.hasam
```

- 5 Create and configure the `SUNW.hasam` resource.

```
clresource create -g SAM-HA -t SUNW.hasam -x QFSName=qfs1,qfs2
-x CatalogFileSystem=/sam_shared sam-ha
```

- 6 Create dependencies between resources within the resource group.

```
clresource set -p Resource_dependencies=sam-hastp qfs-ha
clresource set -p Resource_dependencies=qfs-ha sam-ha
```

- 7 Bring the resource group online.

```
clresourcegroup manage SAM-HA
clresourcegroup online -eM SAM-HA
```

---

**Note** – The Sun QFS file systems must be mounted before bringing the resource group online.

---

**8 Ensure that the resource group is functional on all configured nodes.**

```
clresourcegroup switch -n scnode-B SAM-HA
clresourcegroup switch -n scnode-A SAM-HA
```

## Using the `samd hasstop` Command

The `samd` command's `hasstop` is specifically for use with HA-SAM. The `samd hasstop` command stops the archiver and stager daemons before stopping `sam-amld` and its children.

The `samd hasstop` command must not be used by an administrator in either a stand-alone or shared SAM-QFS environment. In addition, the normal `samd stop` command must not be used on nodes under HA-SAM control.

## Using StorageTek Libraries with HA-SAM

If you are using StorageTek libraries in an HA-SAM environment, the STK daemons need to be notified to use a forced dismount instead of a normal dismount in the case of an involuntary failover. To accomplish this, the `/var/run/hasam_running` file is created on the active Oracle Solaris Cluster node when the HA-SAM resource is brought online. This is a zero-byte file with root permissions. If this file is detected on a node, `sam-stkd` and `sam-stk_helper` use `force_media` to dismount the tape on the drive. This action allows the successful initialization of the library and the drives after an involuntary failover. The `hasam_running` file is deleted when a SAM-QFS resource is brought offline on a node.

In order to ensure the correct failover behavior for StorageTek libraries, make sure the `/var/run/hasam_running` is not deleted if found on an Oracle Solaris Cluster node that has an HA-SAM resource online.



# Configuring Clients Outside of the Cluster

---

This chapter provides information about configuring a Sun QFS shared file system operating in an Oracle Solaris Cluster environment to support automated metadata server failover while providing file system access for clients that are not members of the Oracle Solaris Cluster configuration. This feature is commonly referred to as “Clients Outside of The Cluster” (COTC).

---

**Note** – Before performing any COTC tasks, make sure that you are familiar with information about shared and local host tables.

---

## COTC Task Map

Use the tasks in the following table to configure COTC.

| Step | Task                                                                                        | Description                                                                                                                                   |
|------|---------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | Verify prerequisites and identify clients that are to be configured outside of the cluster. | See “ <a href="#">Configuration Requirements</a> ” on page 42.                                                                                |
| 2    | Install the Oracle Solaris Cluster software.                                                | If not already done, install the Oracle Solaris Cluster software as explained in <i>Oracle Solaris Cluster Software Installation Guide</i> .  |
| 3    | Install the Sun QFS software on the metadata server and metadata failover clients.          | See Chapter 5, “ <a href="#">Installing Sun QFS and SAM-QFS</a> ,” in <i>Sun QFS and Sun Storage Archive Manager 5.3 Installation Guide</i> . |
| 4    | Configure the metadata server.                                                              | See “ <a href="#">How to Configure the Metadata Server</a> ” on page 42.                                                                      |
| 5    | Configure the clients.                                                                      | See “ <a href="#">How to Configure the File System Clients</a> ” on page 46.                                                                  |

# Configuration Requirements

The following table identifies specific configuration rules for each segment of the COTC environment.

| Segment                        | Configuration Rules                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Sun QFS metadata server        | <ul style="list-style-type: none"> <li>■ mm devices must be on a separate physical LUN from data devices.</li> <li>■ mr devices must be on a separate physical LUN from metadata devices.</li> <li>■ mm devices must not have the Oracle Solaris Cluster <code>localonly</code> flag.</li> </ul>                                                                                                                                                                                                                         |
| Sun QFS clients                | <ul style="list-style-type: none"> <li>■ mm devices must be identified as <code>nodev</code> in the <code>mc f</code> file.</li> <li>■ mr devices must follow the <code>/dev/dsk</code> syntax rather than the <code>/dev/did/dsk</code> syntax.</li> <li>■ The definition for the mr device on the client <code>/dev/sdk</code> must match the definition for the <code>/dev/did/dsk</code> on the metadata server.</li> </ul>                                                                                          |
| Oracle Solaris Cluster devices | Set <code>localonly</code> flag on the mr devices.                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Private metadata network       | <ul style="list-style-type: none"> <li>■ Oracle Solaris Cluster nodes that serves as metadata servers should use IPMP groups.</li> <li>■ Oracle Solaris Cluster Sun QFS resource groups should have a logical host name configured for metadata communication between the metadata server and the clients that are outside of the cluster.</li> <li>■ Oracle Solaris Cluster Sun QFS resource groups should be defined to ensure that Sun QFS file system resources depend on the logical host name resource.</li> </ul> |

## ▼ How to Configure the Metadata Server

The following procedure includes embedded examples to better explain what you should expect to see in a COTC configuration.

## 1 Use the `cluster show -t device` command to identify the devices that will be used for the Sun QFS configuration and build the `mcf` file.

Note the use of the `/dev/did` syntax in the following sample `mcf` file for this configuration.

```
#
File system Qfs1
#
Qfs1 2 ma Qfs1 on shared
/dev/did/dsk/d7s0 20 mm Qfs1 on
/dev/did/dsk/d8s0 21 mm Qfs1 on
/dev/did/dsk/d16s0 22 mr Qfs1 on
/dev/did/dsk/d10s0 23 mr Qfs1 on
/dev/did/dsk/d13s0 24 mr Qfs1 on
#
File system Qfs2
#
Qfs2 5 ma Qfs2 on shared
/dev/did/dsk/d9s0 50 mm Qfs2 on
/dev/did/dsk/d11s0 51 mm Qfs2 on
/dev/did/dsk/d17s0 52 mr Qfs2 on
/dev/did/dsk/d12s0 53 mr Qfs2 on
/dev/did/dsk/d14s0 54 mr Qfs2 on
/dev/did/dsk/d15s0 55 mr Qfs2 on
/dev/did/dsk/d18s0 56 mr Qfs2 on
```

Example `/etc/opt/SUNWsamfs/defaults.conf` file:

```
trace
all = on
sam-fsd.size = 10M
sam-sharefsd.size = 10M
endtrace
```

Example `/etc/opt/SUNWsamfs/samfs.cmd` file:

```
fs = Qfs1
meta_timeo=0

fs = Qfs2
meta_timeo=0
```

## 2 Build the Sun QFS file system hosts tables.

The example configuration uses shared and local file system host files for this QFS shared file system setup.

---

**Note** – To communicate with metadata clients that are outside of the cluster, you must establish Sun QFS metadata traffic over the Sun QFS network. Because the metadata client is not a member of the Oracle Solaris Cluster configuration, the example uses a logical host for this traffic. In the example configuration, `sc-qfs1` is the logical host name.

---

## 3 Build the shared host table on the metadata server.

Use the `clnode show` command and syntax to obtain the host order information to build the table.

For example:

```
ctelab30:root> clnode show
```

```
=== Cluster Nodes ===
```

```
Node Name: ctelab30
Node ID: 1
Enabled: yes
privatehostname: clusternode1-priv
reboot_on_path_failure: disabled
globalzoneshares: 1
defaultpsetmin: 1
quorum_vote: 1
quorum_defaultvote: 1
quorum_resv_key: 0x4AB78CC400000001
Transport Adapter List: e1000g1, e1000g3

Node Name: ctelab31
Node ID: 2
Enabled: yes
privatehostname: clusternode2-priv
reboot_on_path_failure: disabled
globalzoneshares: 1
defaultpsetmin: 1
quorum_vote: 1
quorum_defaultvote: 1
quorum_resv_key: 0x4AB78CC400000002
Transport Adapter List: e1000g1, e1000g3
```

Example of the /etc/opt/SUNWsamfs/hosts.Qfs1 file:

In the following example, each file system must have its own host table.

```
#
#MDS
Shared MDS Host file for family set 'Qfs1'
#
#
ctelab30 clusternode1-priv,sc-qfs1 1 - server
ctelab31 clusternode2-priv,sc-qfs1 2 -
ctelab32 ctelab32,ctelab32-4 - -
ctelab33 ctelab33,ctelab33-4 - -
ctelab28 ctelab28,ctelab28-4 - -
```

Example of the /etc/opt/SUNWsamfs/hosts.Qfs1.local file:

In the following example, each file system must have its own host table.

```
#
#MDS
Local MDS Host file for family set 'Qfs1'
#
#
ctelab30 clusternode1-priv
ctelab31 clusternode2-priv
```

**4 Create the file system by running `sammkfs` on each file system.**

```
/opt/SUNWsamfs/sbin/sammkfs -S Qfs1
```

**5 Add file system mounts to the `/etc/vfstab` file.**

Edit the `/etc/vfstab` file on each cluster node and append the Sun QFS file system mounts.

```
###
QFS Filesystems
###
Qfs1 - /cluster/qfs1 samfs - no shared
Qfs2 - /cluster/qfs2 samfs - no shared
```

**6 Mount the file systems.****a. Make the mount points on each cluster node.**

```
mkdir -p /cluster/qfs1 /cluster/qfs2
```

**b. Mount `Qfs1` and `Qfs2` on each cluster node.****7 Create the Oracle Solaris Cluster metadata server resource group.**

When you have configured Oracle Solaris Cluster and Sun QFS, you need to create the MDS resource group under Oracle Solaris Cluster. Follow these steps:

**a. Add the Sun QFS resource type.**

```
clresourcetype register SUNW.qfs
```

**b. Create a resource group for the metadata server.**

```
clresourcegroup create -n ctelab30,ctelab31 sc-qfs-rg
```

**c. Add the logical host name to the resource group that you just created.**

```
clreslogicalhostname create -g sc-qfs-rg \\
-N qfs_ipmp1@ctelab30,qfs_ipmp1@ctelab31 sc-qfs1
```

**d. Add Sun QFS file system resources to the MDS resource group.**

```
clresource create -g sc-qfs-rg -t SUNW.qfs \\
-x QFSFileSystem=/cluster/qfs1,/cluster/qfs2 -y \\
Resource_dependencies=sc-qfs1 \\
fs-qfs-rs
```

**e. Bring the resource group online.**

```
clresourcegroup online -emM sc-qfs-rg
```

**f. Check resource group status.**

```
clresourcegroup status
```

## 8 Set local mode on the Sun QFS data devices.

For each data device (mr device) that is part of the file system, set local mode. In the following example, each data device in the Qfs1 file system must have local mode set.

```
/usr/cluster/bin/scconf -r -D name=dsk/d16,nodelist=ctelab31
/usr/cluster/bin/scconf -c -D name=dsk/d16,localonly=true
/usr/cluster/bin/scconf -r -D name=dsk/d10,nodelist=ctelab31
/usr/cluster/bin/scconf -c -D name=dsk/d10,localonly=true
/usr/cluster/bin/scconf -r -D name=dsk/d13,nodelist=ctelab31
/usr/cluster/bin/scconf -c -D name=dsk/d13,localonly=true
```

## ▼ How to Configure the File System Clients

After the operating system and Oracle Solaris Cluster software have been installed on the clients, install and configure the Sun QFS software on each metadata client.

### 1 Install the metadata client.

Specific steps differ slightly, depending on the client architecture.

For Oracle Solaris systems:

- Go to the location of the Sun QFS software packages. For example:  

```
cd /net/eyelid/builds/src_ctl/R4_5/6/pkg/obj/SunOS_5.9_sparc
```
- Add the packages.  

```
pkgadd-d . SUNWqfsr SUNWqfsu
```
- If necessary, go to the Oracle eDelivery site and download any additional patches.
- If necessary, apply patches.

For Linux systems:

Follow similar steps to the Oracle Solaris process to install the base packages and apply any patches. Once the install completes, the install process will detect and build the `mc f` file configuration.

- Create mount directories and mount the packages as needed. For example:  

```
mkdir /mnt/pkg /mnt/pkg
mount eyelid-mn.central:/builds/src_ctl/R4_5/6/pkg-linux /mnt/pkg
mount -o loop,ro iso_open.iso /mnt/pkg
```
- Add the packages.  

```
/mnt/pkg/Install
```
- If necessary, go to My Oracle Support (MOS) and download any additional patches.
- If necessary, apply patches.

### 2 Verify MPXIO has been enabled and clients can discover QFS devices.

### 3 Identify devices.

Using the Oracle Solaris `format` command on the clients and the `cluster show -t device` command on the metadata server, match device IDs that identify the devices that will be used for the Sun QFS configuration and build the `mcf` file on the SPARC clients.

Example `mcf` file

```
#
File system Qfs1
#
Qfs1 2 ma Qfs1 on shared
nodev 20 mm Qfs1 on
nodev 21 mm Qfs1 on
/dev/dsk/c6t600C0FF00000000000332B21D0B9000d0s0 22 mr Qfs1 on
/dev/dsk/c6t600C0FF00000000000876E9124FAF9C00d0s0 23 mr Qfs1 on
/dev/dsk/c6t600C0FF000000000004CAD7CC3CDE500d0s0 24 mr Qfs1 on
#
File system Qfs2
#
Qfs2 5 ma Qfs2 on shared
nodev 50 mm Qfs2 on
nodev 51 mm Qfs2 on
/dev/dsk/c6t600C0FF00000000000332B057D2FF100d0s0 52 mr Qfs2 on
/dev/dsk/c6t600C0FF00000000000876E975EDA6A000d0s0 53 mr Qfs2 on
/dev/dsk/c6t600C0FF00000000000876E9780ECA8100d0s0 54 mr Qfs2 on
/dev/dsk/c6t600C0FF000000000004CAD139A855500d0s0 55 mr Qfs2 on
/dev/dsk/c6t600C0FF000000000004CAD4C40941C00d0s0 56 mr Qfs2 on
```

### 4 Configure the `/etc/opt/SUNWsamfs/default.s.conf` file on SPARC and Linux clients.

```
trace
all = on
sam-fsd.size = 10M
sam-sharefsd.size = 10M
endtrace
```

### 5 Configure the `/etc/opt/SUNWsamfs/samfs.cmd` file on SPARC and Linux clients.

```
fs = Qfs1
meta_timeo=0

fs = Qfs2
meta_timeo=0
```

### 6 Build the Sun QFS file system host tables.

Using information already configured on the metadata server, build host tables. Follow the examples below.

---

**Note** – Remember that for metadata communications between the metadata server and metadata clients, the clients that are not members of the cluster must communicate over the logical host.

---

Example of the `/etc/opt/SUNWsamfs/hosts.Qfs1` file:

```
#
#MDC
Shared Client Host file for family set 'Qfs1'
#
#
ctelab30 sc-qfs1 1 - server
ctelab31 sc-qfs1 2 -
ctelab32 ctelab32,ctelab32-4 - -
ctelab33 ctelab33,ctelab33-4 - -
ctelab28 ctelab28,ctelab28-4 - -
```

## 7 Bind each local host to the metadata server.

Edit the local hosts file on each client to bind the file system to the metadata server. The `/etc/opt/SUNWsamfs/hosts.fsname`. local information will be different for each client. In the following example, the client is configured to use the interface on `ctelab32-4` to bind to host `sc-qfs1` for metadata traffic.

```
#
#MDC
Local Client Host file for family set 'Qfs1'
#
#
ctelab30 sc-qfs1@ctelab32-4
ctelab31 sc-qfs1@ctelab32-4
```

## 8 In the `/etc/vfstab` file on each client node, append the Sun QFS file system mounts.

SPARC: Example of the `/etc/vfstab` file

```
###
QFS Filesystems
###
Qfs1 - /cluster/qfs1 samfs - no shared
Qfs2 - /cluster/qfs2 samfs - no shared
```

Linux: Example of the `/etc/vfstab` file

```
###
QFS Filesystems
#
Qfs1 /cluster/qfs1 samfs noauto,rw,shared 0 0
Qfs2 /cluster/qfs2 samfs noauto,rw,shared 0 0
```

## 9 Mount file systems on each client.

### a. Make the mount points on each client.

```
mkdir -p /cluster/qfs1 /cluster/qfs2
```

### b. Mount the file systems on each cluster node.

```
mount Qfs1;mount Qfs2
```

## Configuration Example

The following example configuration uses shared and local file system host files for this Sun QFS shared file system setup. To communicate with metadata clients that are outside the Oracle Solaris Cluster, you must establish the Sun QFS metadata traffic over the Sun QFS network. Because the metadata client is not a member of the Oracle Solaris Cluster configuration, use a logical host for this traffic. In the example configuration, `sc-qfs1` is this host name.

The following example configuration files show a tested configuration that consists of four SPARC nodes and one AMD x64 node. These nodes are identified as follows:

| Name     | Role            | Architecture | Notes                       |
|----------|-----------------|--------------|-----------------------------|
| ctelab30 | Metadata server | SPARC        | Oracle Solaris Cluster Node |
| ctelab31 | Metadata server | SPARC        | Oracle Solaris Cluster Node |
| ctelab32 | Metadata client | SPARC        | Sun QFS Client Node         |
| ctelab33 | Metadata client | SPARC        | Sun QFS Client Node         |
| ctelab28 | Metadata client | AMD64        | Sun QFS Linux Client Node   |

Sample `/etc/hosts` File for All Nodes

Prepare `/etc/hosts` file on all nodes to append settings to existing `/etc/hosts` table.

```
SC Cluster Nodes
129.152.4.57 ctelab30 # Cluster Node
129.152.4.58 ctelab31 # Cluster Node
129.152.4.59 ctelab32 # QFS Client Node
129.152.4.60 ctelab33 # QFS Client Node
129.152.4.55 ctelab28 # QFS Client Node
SC Logical
192.168.4.100 sc-qfs1
QFS NET
ctelab30
192.168.4.20 ctelab30-4
192.168.4.160 ctelab30-qfe1-test
192.168.4.210 ctelab30-qfe2-test
192.168.4.60 ctelab30-qfs
ctelab31
192.168.4.21 ctelab31-4
192.168.4.161 ctelab31-qfe1-test
192.168.4.211 ctelab31-qfe2-test
192.168.4.61 ctelab31-qfs
ctelab32
192.168.4.22 ctelab32-qfs
ctelab33
192.168.4.23 ctelab33-qfs
ctelab28
192.168.4.18 ctelab28-qfs
```

## Sample Configuration Files for Metadata Server

### **/etc/nsswitch.conf File**

Ensure that the `/etc/nsswitch.conf` includes cluster. For example:

```
hosts: cluster files dns nis
```

### **/etc/netmasks File**

Add the subnet used in your `/etc/hosts` file to the `/etc/netmasks` file. For example:

```
192.168.4.0 255.255.255.0
```

### **/etc/hostname.qfe1 File**

Identify the NIC port for `qfe1` in the `/etc/hostname.qfe1` file. For example:

```
ctelab30-4 netmask + broadcast + group qfs_ipmp1 up addif
ctelab30-qfe1-test deprecated -failover netmask + broadcast + up
```

### **/etc/hostname.qfe2 File**

Identify the NIC port for `qfe2` in the `/etc/hostname.qfe2` file. For example:

```
ctelab30-qfe2-test netmask + broadcast + deprecated group
qfs_ipmp1 -failover standby up
```

## Client Network Configuration

In this example, both SPARC and AMD-64 clients are used. SPARC clients run the Oracle Solaris OS and AMD-64 clients use the Linux OS.

### **SPARC Oracle Solaris Client Network Configuration**

- `qfe0` used for Public network (129.152.4.0)
- `qfe1` used for Sun QFS private network. (192.168.4.0)

### **AMD-64 RedHat Linux Client Network Configuration**

- `bge0` used for Public network (129.152.4.0)
- `bge1` used for Sun QFS private network. (192.168.4.0)

---

**Note** – On the Linux OS, the NIC ports identified above are hardware address IDs and the OS identifies these as `eth0`, `eth1` at the OS layer.

---

### **/etc/nsswitch.conf File (SPARC and Linux Clients)**

For the clients, make sure that cluster is not listed in the `/etc/nsswitch.conf` file. For example:

```
hosts: files dns nis
```

### **/etc/hostname.qfe1 File (SPARC Clients Only)**

This file should contain the following:

```
ctelab32-4
```

### **/etc/sysconfig/network-scripts/ifcfg-eth1 File (Linux Clients Only)**

This file should contain the following:

```
DEVICE=eth1
BOOTPROTO=static
IPADDR=192.168.4.18
NETMASK=255.255.255.0
ONBOOT=yes
TYPE=Ethernet
```

## **Troubleshooting an Issue With Oracle RAC**

*Problem:* If you use Oracle RAC with Solaris Volume Manager for shared file systems, you might see a message that volumes are not available.

*Solution:* In response to this issue, you can configure the `rac_maxdevretry` parameter in the `/etc/opt/SUNWsamfs/defaults.conf` file. By default, the `rac_maxdevretry` variable is set to 150 (5 minutes).

The items to take in account for adjusting this variable include:

- Number of nodes in the cluster
- Number of physical LUNs
- Number of Solaris Volume Manager disk groups
- Speed of array topology 1G/2G/4G

In the following example `/etc/opt/SUNWsamfs/defaults.conf` file, the time is shortened to 90 seconds (3 minutes).

```
trace
all = on
sam-sharefsd.options = all
sam-fsd.size = 10M
sam-sharefsd.size = 10M
endtrace
rac_maxdevretry = 90
```

This change was based on the following environment:

- Nodes - 8
- Physical LUNs - 32 Shared Physical LUNS with 4G Fibre Channel
- Solaris Volume Manager disk groups - 1
- Shared file systems - 13

An example of this setting is included in the `/etc/opt/SUNWsamfs/examples/defaults.conf` file.

For more information about configuring an Oracle database, see *Oracle Solaris Cluster Data Service for Oracle Real Application Clusters*.