



Sun™ SAM-Remote Administrator's Guide

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054 U.S.A.
650-960-1300

Part No. 816-2094-11
February 2003, Revision 01

Send comments about this document to: docfeedback@sun.com

Copyright 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved.

Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. or other countries.

This document and the product to which it pertains are distributed under licenses restricting their use, copying, distribution, and decompilation. No part of the product or of this document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and in other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, AnswerBook2, docs.sun.com, Solaris, SunOS, and Sun StorEdge are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and in other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and in other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc. Adobe is a registered trademark of Adobe Systems, Incorporated.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

U.S. Government Rights—Commercial use. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés.

Sun Microsystems, Inc. a les droits de propriété intellectuels relatants à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et sans la limitation, ces droits de propriété intellectuels peuvent inclure un ou plus des brevets américains énumérés à <http://www.sun.com/patents> et un ou les brevets plus supplémentaires ou les applications de brevet en attente dans les Etats-Unis et dans les autres pays.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, AnswerBook2, docs.sun.com, Solaris, SunOS, et Sun StorEdge sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. Adobe est une marque SPARC enregistrée de Adobe Systems, Incorporated.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

LA DOCUMENTATION EST FOURNIE "EN L'ÉTAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISÉE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.



Contents

How This Book Is Organized	v
Using UNIX Commands	vi
Typographic Conventions	vi
Shell Prompts	vii
Related Documentation	viii
Accessing Sun Documentation Online	viii
▼ To View the Sun QFS, Sun SAM-FS, and Sun SAM-QFS Software Documents Online	ix
Licensing	ix
Installation Assistance	ix
Sun Welcomes Your Comments	x
1. Sun SAM-Remote Overview	1
Features	1
Requirements	2
Limitations	3
Technical Overview	3
Sun SAM-Remote Server Overview	4
Sun SAM-Remote Client Overview	5
Interaction Between the Sun SAM-Remote Server and the Sun SAM-Remote Client	5

Library Catalogs 5

Archiving 6

2. Configuring Sun SAM-Remote Software 7

Example Configuration 7

Configuring the Software 8

- ▼ To Log In to the Potential Server and Client Hosts 9
- ▼ To Verify Client and Server Configurations 9
- ▼ To Edit the `mcf` Files 11
- ▼ To Define a Sun SAM-Remote Client 13
- ▼ To Define a Sun SAM-Remote Server in the Server's `mcf` File 14
- ▼ To Create the Sun SAM-Remote Server Configuration File 15
- ▼ To Enable Archiving 18

3. Recycling With Sun SAM-Remote 23

Recycling in a Sun SAM-Remote Environment—Method 1 24

Configuration Files for Server `sky` 25

Configuration Files for Client `zeke` 26

- ▼ To Configure Recycling—Method 1 27

Recycling in a Sun SAM-Remote Environment—Method 2 49

- ▼ To Configure Recycling—Method 2 49

Glossary 51

Index 63

Preface

This manual, the *Sun SAM-Remote Administrator's Guide*, describes the Sun™ SAM-Remote client and the Sun SAM-Remote server storage management system. This software allows you to share libraries and other removable media devices in a Sun SAM-FS or Sun SAM-QFS environment. All host systems included in a Sun SAM-Remote environment must have the same Sun SAM-FS or Sun SAM-QFS software release level installed and operational.

The *Sun SAM-Remote Administrator's Guide* is written for system administrators who are responsible for installing, configuring, and maintaining Sun SAM-FS and Sun SAM-QFS software. You, the system administrator, are assumed to be knowledgeable about Sun Solaris™ operating environment procedures, including creating accounts, performing system backups, and other basic Sun Solaris system administrator tasks. It is also assumed that you are familiar with installing, configuring, and using Sun SAM-FS or Sun SAM-QFS software.

How This Book Is Organized

This manual contains the following chapters:

Chapter 1 is an overview.

Chapter 2 explains the configuration procedure.

Chapter 3 explains how to recycle with Sun SAM-Remote software.

The glossary defines terms used in this and other Sun QFS, Sun SAM-FS, and Sun SAM-QFS documentation.

Using UNIX Commands

This document might not contain information on basic UNIX® commands and procedures such as shutting down the system, booting the system, and configuring devices.

See one or more of the following for this information:

- *Solaris Handbook for Sun Peripherals*
- AnswerBook2™ online documentation for the Solaris operating environment
- Other software documentation that you received with your system

Typographic Conventions

TABLE P-1 lists the typographical conventions used in this manual.

TABLE P-1 Typographical Conventions

Typeface	Meaning	Examples
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. % You have mail.
AaBbCc123	What you type, when contrasted with on-screen computer output	% su Password:
<i>AaBbCc123</i>	Book titles, new words or terms, words to be emphasized. Replace command-line variables with real names or values.	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be superuser to do this. To delete a file, type <code>rm filename</code> .

TABLE P-1 Typographical Conventions (*Continued*)

Typeface	Meaning	Examples
[]	In syntax, brackets indicate that an argument is optional.	scmadm [-d <i>sec</i>] [-r <i>n[:n][,n]...</i>] [-z]
{ arg arg }	In syntax, braces and pipes indicate that one of the arguments must be specified.	sndradm -b { <i>phost</i> <i>shost</i> }
\	At the end of a command line, the backslash (\) indicates that the command continues on the next line.	atm90 /dev/md/rdisk/d5 \ /dev/md/rdisk/d1

Shell Prompts

TABLE P-2 shows the shell prompts that this manual uses.

TABLE P-2 Shell Prompts

Shell	Prompt
C shell	<i>machine-name%</i>
C shell superuser	<i>machine-name#</i>
Bourne shell and Korn shell	\$
Bourne shell and Korn shell superuser	#

Related Documentation

This manual is part of a set of documents that describes the operations of the Sun QFS, Sun SAM-FS, and Sun SAM-QFS software products. TABLE P-3 shows the complete documentation set for these products.

TABLE P-3 Related Documentation

Title	Part Number
<i>Sun SAM-Remote Administrator's Guide</i>	816-2094
<i>Sun QFS, Sun SAM-FS, and Sun SAM-QFS Disaster Recovery Guide</i>	816-2540
<i>Sun QFS, Sun SAM-FS, and Sun SAM-QFS File System Administrator's Guide</i>	816-2542
<i>Sun QFS, Sun SAM-FS, and Sun SAM-QFS Installation and Configuration Guide</i>	816-2543
<i>Sun SAM-FS and Sun SAM-QFS Storage and Archive Management Guide</i>	816-2544

Accessing Sun Documentation Online

The Sun QFS, Sun SAM-FS, and Sun SAM-QFS software distribution includes PDF files for the documents for these products. Viewing PDF files requires the Adobe Acrobat Reader software, which is available for free from the following web site:

www.adobe.com/products/acrobat/readerstep.html

PDF files of documents for these and other storage-related products are also posted at Sun's Network Storage documentation web site, which is as follows:

www.sun.com/products-n-solutions/hardware/docs/Network_Storage_Solutions

For specific instructions on accessing the Sun QFS, Sun SAM-FS, and Sun SAM-QFS documents, see "To View the Sun QFS, Sun SAM-FS, and Sun SAM-QFS Software Documents Online" on page ix.

A complete set of Solaris documentation and many other titles are located at the following URL:

`docs.sun.com`

▼ To View the Sun QFS, Sun SAM-FS, and Sun SAM-QFS Software Documents Online

1. Go to the Storage Software web page:

This is located at the following URL:

`www.sun.com/products-n-solutions/hardware/docs/Software/Storage_Software`

2. Click on the appropriate link.

The links are as follows:

- Sun StorEdge Performance Suite (Sun QFS software)
- Sun StorEdge Utilization Suite (Sun SAM-FS software)

Licensing

For information about obtaining licenses for Sun QFS, Sun SAM-FS, or Sun SAM-QFS software, contact your Sun sales representative or your authorized service provider (ASP).

Installation Assistance

The Sun Enterprise Services group provides installation and configuration services. To obtain these services within the United States, contact Sun's Enterprise Services group at 1-800-USA4SUN. Outside the United States, contact your local Enterprise Services sales representative.

Sun Welcomes Your Comments

Sun is interested in improving its documentation and welcomes your comments and suggestions. You can email your comments to Sun at the following address:

`docfeedback@sun.com`

Please include the part number (816-2094-11) of your document in the subject line of your email.

Sun SAM-Remote Overview

The Sun SAM-Remote client and the Sun SAM-Remote server form a client/server implementation that enables libraries and other removable media devices to be shared between Sun SAM-FS and Sun SAM-QFS host systems. Sun SAM-Remote enables you to configure multiple storage clients that archive and stage files from a centralized tape library or magneto-optical library. For example, if you have host systems on a network that spans a large geographical area, files created in one city can be archived to cartridges in a library located miles away.

This chapter contains the following sections:

- “Features” on page 1
- “Requirements” on page 2
- “Limitations” on page 3
- “Technical Overview” on page 3

Features

Sun SAM-Remote software provides the following advantages:

- Enables you to configure remote sharing of an expensive removable media resource, such as a library, between one or more Sun SAM-Remote clients.
- Enables clients to migrate data to a server.
- Enables multiple Sun SAM-FS or Sun SAM-QFS servers to be hosts to one another. In a Sun SAM-Remote environment, the server is the host system that is configured with an equipment type of `ss` in the `mcf` file.

You can configure the Sun SAM-Remote server and clients to provide multiple archive copies between two or more Sun Solaris host systems. For example, you can configure two Solaris systems running Sun SAM-FS software as both a Sun SAM-Remote server and a Sun SAM-Remote client to each other. Benefits of this

configuration include the ability to create local copies for each server with an additional archive copy of data on the other server. File systems can be shared between servers using standard NFS. In the event of a loss of access to the local library, Sun SAM-Remote software would automatically retrieve file data from the archive copy. Users of both servers would have uninterrupted access to their data, even if their primary storage library were unavailable.

FIGURE 1-1 shows an environment configured with two Sun SAM-Remote host system servers. Each has two clients.

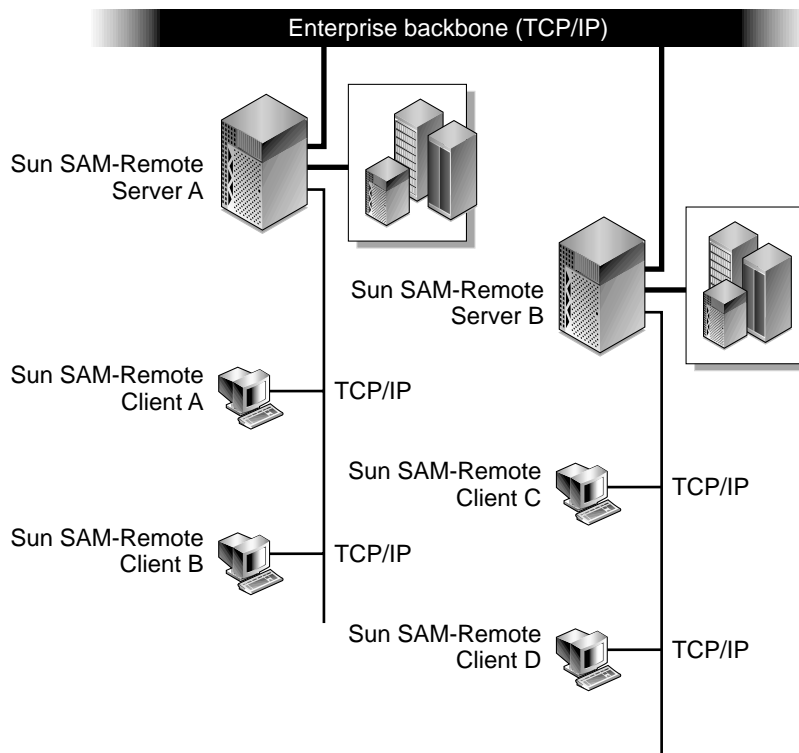


FIGURE 1-1 Sun SAM-Remote Servers and Clients

Requirements

Before attempting to configure a Sun SAM-Remote 4.0 environment, make sure that your environment includes the following software and hardware:

- SPARC^R systems with licensed, installed, and operable Sun SAM-FS or Sun SAM-QFS 4.0 storage and archive management software packages.
- Host systems with identical Sun SAM-FS or Sun SAM-QFS software revision levels and identical patch collections installed. If some host systems have to be upgraded, you can find information about that topic in the *Sun QFS, Sun SAM-FS, and Sun SAM-QFS Installation and Configuration Guide*.
- One host system to act as the Sun SAM-Remote server with at least one Sun SAM-FS or Sun SAM-QFS file system installed upon it.
- A network connection running a TCP/IP connection between the clients and the server upon which the Sun SAM-FS or Sun SAM-QFS 4.0 software is installed.

Limitations

The storage and archive manager treats cartridges in a remote library no differently from the way it treats cartridges in a local library. The following information, however, indicates the limits of Sun SAM-Remote software:

- You can recycle media using Sun SAM-Remote, but you should attempt this only after thoroughly testing your environment. For more information, see “Recycling With Sun SAM-Remote” on page 23.
- You can configure up to 10 clients per Sun SAM-Remote server.
- Only one daemon on a Sun SAM-Remote client can communicate to the Sun SAM-Remote server.
- Host systems that are also included in a Sun QFS shared file system cannot be configured in a Sun SAM-Remote environment.

Technical Overview

The Sun SAM-Remote clients interact with the Sun SAM-Remote server using a TCP/IP connection. The network between the Sun SAM-Remote clients can be any network type supported by the Sun Solaris operating environment, such as Ethernet, Fast Ethernet, FDDI, Fiber Channel, and HIPPI.

FIGURE 1-2 shows Sun SAM-Remote client and Sun SAM-Remote server interactions.

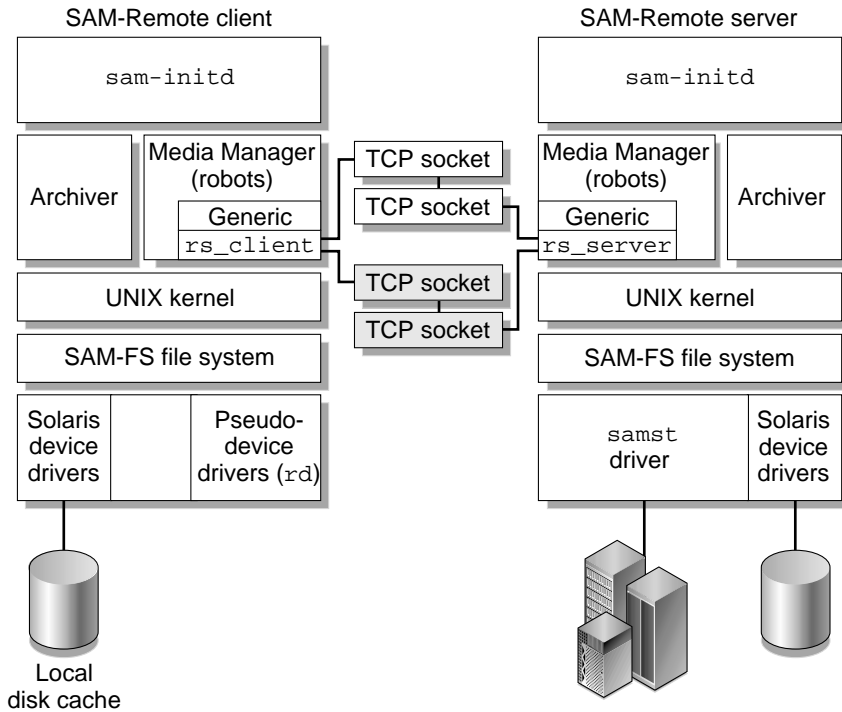


FIGURE 1-2 Sun SAM-Remote Server and Client Interactions

Sun SAM-Remote Server Overview

The Sun SAM-Remote server consists of a full-capability Sun SAM-FS or Sun SAM-QFS storage management host and a Sun SAM-Remote server daemon that defines libraries to be shared among the clients. At least one Sun SAM-FS or Sun SAM-QFS file system must be configured on the Sun SAM-Remote server.

You define a host system as a Sun SAM-Remote server by adding a line in the server system's `/etc/opt/SUNWsamfs/mcf` file with an equipment type of `ss`. You must provide a unique family set name for each server. Up to 10 clients can be configured per server. For more information about the server daemon, see the `sam-remote(7)` man page.

Sun SAM-Remote Client Overview

The Sun SAM-Remote client is a Sun SAM-FS or Sun SAM-QFS host system that establishes a Sun SAM-Remote client daemon containing a number of pseudo-devices.

You define a host system as a Sun SAM-Remote client by adding a line in the client system's `/etc/opt/SUNWsamfs/mcf` file with an equipment type of `sc`. For more information about the client daemon, see the `sam-remote(7)` man page.

A pseudo-device defines a network connection to an actual removable media device on the Sun SAM-Remote server. Pseudo-devices have an equipment type of `rd`, which is a mnemonic for *remote device*. You define the pseudo-devices in the Sun SAM-Remote client's `/etc/opt/SUNWsamfs/mcf` file. The Sun SAM-Remote daemon and pseudo-devices are associated with one particular server.

By default, the Sun SAM-Remote daemon supports up to 32 pseudo-devices for each client. The actual number of pseudo-devices to be used by the client is configurable. When determining how many pseudo-devices should be configured per client, think of these devices as the number of simultaneous data transfers that can occur between the client and the server. As more pseudo-devices are defined, the possibility of increasing the total network traffic load increases. It is up to you, the system administrator, to determine the actual number of pseudo-devices needed for the system.

Interaction Between the Sun SAM-Remote Server and the Sun SAM-Remote Client

The Sun SAM-Remote server daemon, `sam-serverd`, listens for the clients on port 1000. You can configure a different port in the Sun Solaris `/etc/services` directory with a service name of `rmtsam`. When a Sun SAM-Remote client connects to the Sun SAM-Remote server, the `sam-serverd` daemon establishes a connection on another port and communicates this port number to that client, using the defined port. The socket size is passed to the client. The socket size is configurable and is described in more detail in the “Configuring Sun SAM-Remote Software” on page 7.

Library Catalogs

The Sun SAM-Remote library catalog is a subset of the catalog located on the Sun SAM-Remote server. The client catalog is updated in real time. The slots allotted to a Sun SAM-Remote client catalog are controlled only by the Sun SAM-Remote server.

Upon initialization, the system builds a client catalog and passes it to the Sun SAM-Remote client based on information from the Sun SAM-Remote server catalog file. After the connection between the host and client is established, media available to the client is flagged as available. If the connection between the client and server is lost, the media on the client side is flagged as unavailable. You can view the media availability through the `samu(1M) v` display. The information that appears in the `samu(1M) v` display on the client is a subset of that which appears in the `v` display on the server. You should typically access the media catalog through the `samu(1M) v` display on Sun SAM-Remote server. For more information about the Sun SAM-Remote server client file, see “Configuring Sun SAM-Remote Software” on page 7.

Changes to the catalog are passed between hosts as necessary. Any changes in the server catalog that involve a media type associated with a client are passed to the client, and the client catalog is updated.

Archiving

Sun SAM-Remote archive processing is the same as Sun SAM-FS and Sun SAM-QFS archive processing. The Sun SAM-Remote client makes a mount request to be added to the server’s mount request table. The client then waits for the server to respond with a message indicating that the media is mounted. Archiving begins when the media is available.

Configuring Sun SAM-Remote Software

This chapter explains how to perform an initial configuration of the Sun SAM-Remote server and client software. It includes the following sections:

- “Example Configuration” on page 7
- “Configuring the Software” on page 8

Example Configuration

FIGURE 2-1 depicts the sample configuration used in this chapter’s procedures. The examples in this chapter show how to configure a Sun SAM-Remote server called `chicago`.

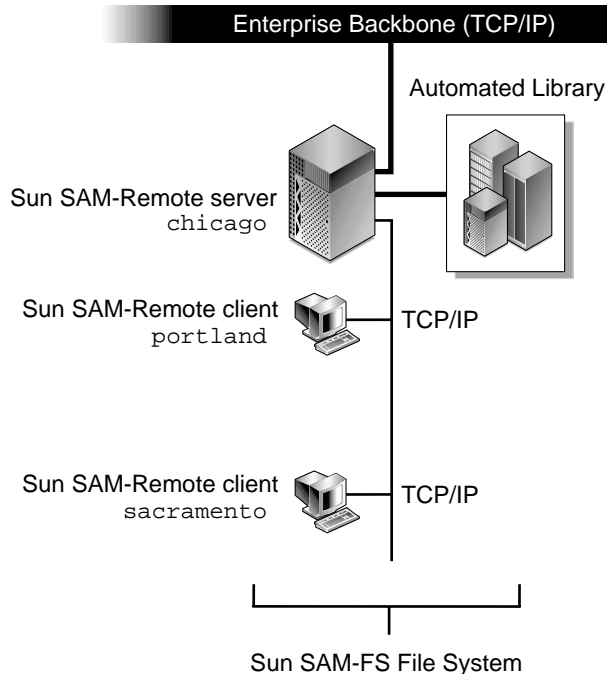


FIGURE 2-1 Example Sun SAM-Remote Configuration

The Sun SAM-FS file systems on `portland` and `sacramento` use `chicago` as their Sun SAM-Remote server.

In the examples in this chapter, the Sun SAM-FS file systems write some of their archive copies to cartridges controlled by `chicago`.

Configuring the Software

The following procedures explain how to configure the Sun SAM-Remote software on a Sun SAM-Remote server and on one or more Sun SAM-Remote clients. These procedures must be performed in the order shown, which is as follows:

1. “To Log In to the Potential Server and Client Hosts” on page 9
2. “To Verify Client and Server Configurations” on page 9
3. “To Edit the `mcf` Files” on page 11

4. “To Define a Sun SAM-Remote Client” on page 13
5. “To Define a Sun SAM-Remote Server in the Server’s mcf File” on page 14
6. “To Create the Sun SAM-Remote Server Configuration File” on page 15
7. “To Enable Archiving” on page 18

In the following steps, you log in to the host systems, verify existing software revision levels, and upgrade the software as needed.

▼ To Log In to the Potential Server and Client Hosts

You must log in to all potential server and client hosts as the superuser.

1. Log in to the Sun SAM-Remote server as the superuser.

You must have superuser access to the server system on which the Sun Sun SAM-Remote software is to be installed.

2. Log in to the Sun SAM-Remote client(s) as the superuser.

You must have superuser access to the client system or systems on which the Sun SAM-Remote software is to be installed.

▼ To Verify Client and Server Configurations

The following steps ensure that you have the required software levels installed on the systems to be configured as part of a Sun SAM-Remote environment.

1. Issue the `pkginfo(1M)` command with its `-l` option on all hosts to be configured as a Sun SAM-Remote client or server.

You must have the same release and revision level of Sun SAM-FS or Sun SAM-QFS software level installed on all client and server hosts to be configured as part of a Sun SAM-Remote environment. For example:

CODE EXAMPLE 2-1 Using `pkginfo(1)`

```
portland# pkginfo -l SUNWsamfs
PKGINST:  SUNWsamfs
NAME:     Sun SAM-FS and Sun SAM-QFS software Solaris 2.8
CATEGORY: system
ARCH:    sparc
```

CODE EXAMPLE 2-1 Using `pkginfo(1)` (Continued)

```
VERSION: 4.0.5,REV=5.8.2003.01.12
VENDOR: Sun Microsystems, Inc.
PSTAMP: boomerang-20020712183351
INSTDATE: Jan 20 2003 07:30
HOTLINE: Please contact your local service provider
STATUS: completely installed
FILES: 489 installed pathnames
      12 shared pathnames
      1 linked files
      51 directories
      179 executables
      35813 blocks used (approx)

portland#
```

2. Examine the output from the `pkginfo(1)` command.

Using the example output shown in CODE EXAMPLE 2-1, you can see that the server is running software version 4.0.5, and any systems included in an environment with this server would also have to be running 4.0.5.

It is assumed that the Sun SAM-FS or Sun SAM-QFS environments are properly configured and operational.

3. Issue the `showrev(1M)` command with its `-p` option on all hosts to be configured as a Sun SAM-Remote client or server.

You must have the same patch collection installed on all client and server hosts to be configured as part of the Sun SAM-Remote environment. For example:

CODE EXAMPLE 2-2 Using `showrev(1M)`

```
portland# showrev -p | grep SUNWsamfs
Patch: 113546-07 Obsoletes: Requires: Incompatibles: Packages:
SUNWsamfs
portland#
```

4. Examine the output from the `showrev(1M)` command.

Using the example output shown in CODE EXAMPLE 2-2, you can see that the server is running patch 113546-07, and any systems included in an environment with this server would also have to be running patch 113546-07.

5. Repeat Step 1, Step 2, Step 3, and Step 4 for each system to be configured in the environment.

6. (Optional) Upgrade the software as necessary.

If the information from the `pkginfo(1)` command reveals that all systems to be included in the Sun SAM-Remote environment are running the same software release level and same patch level, you do not need to perform this step.

If some systems to be configured as part of a Sun SAM-Remote environment are running earlier versions of the software or patches, upgrade all systems to the latest software levels. Using CODE EXAMPLE 2-1 as an example, if you are running a Sun SAM-FS or Sun SAM-QFS version earlier than version 4.0.5 on any system, you must upgrade to 4.0.5.

For information about performing software upgrades, see the *Sun QFS, Sun SAM-FS, and Sun SAM-QFS Installation and Configuration Guide*.

▼ To Edit the `mcf` Files

1. From the Sun SAM-Remote server, stop the Sun SAM-FS or Sun SAM-QFS file system.

If the Sun SAM-FS or Sun SAM-QFS system is running, you must stop it.

- a. Issue the `samcmd(1M)` command with its `idle eq` option to idle the removable media drives under the control of the Sun SAM-FS or Sun SAM-QFS software.

For example:

```
# samcmd idle eq
```

Argument	Definition
<code>eq</code>	The equipment ordinal of the removable media drive being addressed, as defined in the <code>mcf</code> file.

Issue a `samcmd(1M)` command for each removable media drive in the environment. For more information about the `samcmd(1M)` command, see the `samcmd(1M)` man page.

Alternatively, you can also idle the drives by using the `samu(1M)` operator utility or by using either the `robottool(1M)` or `libmgr(1M)` graphical user interface (GUI) tools.

Note – The drives in your Sun SAM-FS or Sun SAM-QFS environment should be idled before you issue the `samd stop` command. This enables the archiver, stager, and other processes to complete current tasks. This also enables the cartridges to be unloaded and put into their storage slots.

b. Issue the `samd(1M)` command with its `stop` option to stop the `sam-initd` daemon and its child processes.

```
# samd stop
```

The `samd(1M)` command is installed in `/opt/SUNWsamfs/sbin`.

2. On a client, use `vi(1)` or another editor to edit the existing Sun SAM-FS or Sun SAM-QFS `/etc/opt/SUNWsamfs/mcf` file.

The goal of this step is to define the host as a Sun SAM-Remote client.

CODE EXAMPLE 2-3 shows the edited `mcf` file on client `portland`. The `mcf` file defines a Sun SAM-FS file system and shows the Sun SAM-Remote client `portland` being defined to the Sun SAM-Remote server `chicago`.

CODE EXAMPLE 2-3 `mcf` File on `portland`

```
# mcf file on portland
#
# Sun SAM-FS file system
#
# Equipment          Eq   Eq  Family    Dev  Additional
# Identifier         Ord  Ty  Set       St   Parameters
# =====          ===  ==  =====  ==  =====
samfs1              1   ms  samfs1    on
/dev/dsk/c1t1d0s0   10  md  samfs1    on  /dev/rdisk/c1t1d0s0
/dev/dsk/c1t2d0s0   12  md  samfs1    on  /dev/rdisk/c1t2d0s0
#
# Define Sun SAM-Remote Client portland to Sun SAM-Remote server chicago
#
/etc/opt/SUNWsamfs/rmt200 200  sc  chicagoss on  /var/opt/SUNWsamfs/catalog/tcat
/dev/samrd/rd0         201  rd  chicagoss on
/dev/samrd/rd1         202  rd  chicagoss on
```

The `mcf` entry on the client consists of a single-line entry for the Sun SAM-Remote client and a pseudo-device entry for each device you want to configure. These entries follow the syntax as defined on the `mcf(4)` man page.

The first set of entries defines a Sun SAM-FS file system.

The second set of entries defines the Sun SAM-Remote client, `portland`, to the Sun SAM-Remote server, `chicago`. The first line defines the Sun SAM-Remote server itself. The fields are as follows:

- The Equipment Identifier field is the path name to the client configuration file, which is created later in “To Define a Sun SAM-Remote Client” on page 13. In this example, the configuration file is named `/etc/opt/SUNWsamfs/rmt200`.

- The Equipment Ordinal field contains a unique number such that $1 < \textit{equipment_ordinal} < 65535$. This equipment ordinal is 200.
- The Equipment Type field contains a two-letter mnemonic, `sc`, which identifies a Sun SAM-Remote client.
- The Family Set field, `chicagoss`, is the same as the family set name of the server. This is the family set name of the daemon to use on this particular server. A Sun SAM-Remote server can have one server daemon per client.
- The Device State field specifies `on`.
- The Additional Parameters field is optional. As shown, a path to the catalog file can be specified here.

The last two entries in this `mcf` file define the Sun SAM-Remote pseudo-devices. A pseudo-device defines a network connection to an actual device on the Sun SAM-Remote server. These entries are as follows:

- The Equipment Identifier field is the path name to the `/dev/samrd/rd*` entry to be used by the pseudo-device. These entries are created when the system is rebooted. You can define up to 32 pseudo-devices.
- The Equipment Type field is the two-letter mnemonic `rd` for pseudo-devices.
- The Family Set field, `chicagoss`, is the same as the family set name of the client entry.

3. (Optional) On additional clients, use `vi(1)` or another editor to edit the existing Sun SAM-FS or Sun SAM-QFS `/etc/opt/SUNWsamfs/mcf` file.

If you have additional clients, you must complete this step for each additional Sun SAM-Remote client. Follow the same procedure outlined in Step 2.

In this chapter's example, the same configuration process must be completed for client `sacramento`. For this system, edit the `mcf` file and copy the last set of lines from `portland`'s `mcf` file to `sacramento`'s `mcf` file. These are the lines that define the host to `chicago` as a Sun SAM-Remote client.

▼ To Define a Sun SAM-Remote Client

The Sun SAM-Remote client's configuration file contains a single-line entry: the name of the Sun SAM-Remote server. As shown in "To Define a Sun SAM-Remote Client" on page 13 in Step 2, the full path name of this client configuration file is specified in the client's `mcf` file.

1. On the client, use `vi(1)` or another editor to open a file to be known as the Sun SAM-Remote client configuration file.

For example:

```
portland# vi /etc/opt/SUNWsamfs/rmt200
```

2. Edit the file and include only the name of the Sun SAM-Remote server.

The result of this step is a one-line file.

CODE EXAMPLE 2-4 shows the client configuration file on `portland` after you have edited it. It points to the Sun SAM-Remote server called `chicago`.

CODE EXAMPLE 2-4 Client Configuration File

```
portland# cat /etc/opt/SUNWsamfs/rmt200
chicago
```

3. Repeat Step 1 and Step 2 for each Sun SAM-Remote client.

If you have more than one client, create a client file on each client.

▼ To Define a Sun SAM-Remote Server in the Server's `mcf` File

This step defines a Sun SAM-Remote server in the server's `mcf` file.

- **On the Sun SAM-Remote server, use `vi(1)` or another editor to edit the existing Sun SAM-FS or Sun SAM-QFS `/etc/opt/SUNWsamfs/mcf` file to define the system as a Sun SAM-Remote server.**

In this step's example, the `mcf` file on server `chicago` is edited. The resulting `mcf` file defines a Sun SAM-FS file system and also defines `chicago` as a Sun SAM-Remote server.

CODE EXAMPLE 2-5 shows the `mcf` file on `chicago`.

CODE EXAMPLE 2-5 `mcf` File on `chicago`

```
# mcf file on Sun SAM-Remote server chicago:
# Eq Identifier Eq Ord Eq Typ Fam Set Dev St Addl Params
#
samfs1          1    ms    samfs1 on
/dev/dsk/c2t6d0s0 11  md    samfs1 on /dev/rdisk/c2t6d0s0
/dev/dsk/c2t6d0s1 12  md    samfs1 on /dev/rdisk/c2t6d0s1
#
# define a tape library that client portland can use:
```


CODE EXAMPLE 2-5 mcf File on chicago (Continued)

```
/dev/samst/c0t3u0 100 rb rb100 on /var/opt/SUNWsamfs/catalog/rb100.cat
/dev/rmt/0cbn 101 tp rb100 on
/dev/rmt/1cbn 102 tp rb100 on

# Define Sun SAM-Remote server chicago
#
/etc/opt/SUNWsamfs/rmt200 50 ss chicagoss on
```

These entries follow the syntax as defined in `mcf(4)`, and in this example file, they are as follows:

- The Equipment Identifier field is the path name to the server configuration file, which you configure in Step t. In this example, the file is named `/etc/opt/SUNWsamfs/rmt200`.
- The Equipment Ordinal field contains a unique number such that $1 \leq \textit{equipment_ordinal} \leq 65535$. In this example, the equipment ordinal is 50.
- The Equipment Type field contains a two-letter mnemonic, `ss`, that identifies the Sun SAM-Remote Server.
- The Family Set field, `chicagoss`, matches the family set name used in the `mcf` file of the client(s). Note that a Sun SAM-Remote server can have more than one server daemon defined.
- The Device State field, which is optional, specifies `on` in this example.
- The Additional Parameters field is optional.

Note – You must have at least one Sun SAM-FS or Sun SAM-QFS file system configured in the `mcf` file for the Sun SAM-Remote server.

▼ To Create the Sun SAM-Remote Server Configuration File

The Sun SAM-Remote server configuration file defines the disk buffer characteristics and media to be used for each client. Ten clients can be configured per server daemon. If you want to support more clients, you must configure another Sun SAM-Remote server daemon as described previously in “To Edit the `mcf` Files” on page 11 (Step 2) and in “To Define a Sun SAM-Remote Client” on page 13.

1. **On the server, use `vi(1)` or another editor to open a file to be known as the Sun SAM-Remote server configuration file.**
2. **Write the server configuration file.**

CODE EXAMPLE 2-6 shows an example server configuration file, `/etc/opt/SUNWsamfs/rmt200`, which resides on Sun SAM-Remote server `chicago`. This file defines clients `portland` and `sacramento`.

CODE EXAMPLE 2-6 Server Configuration File `rmt200`

```
#
# Sun SAM-Remote server config file /etc/opt/SUNWsamfs/rmt200
#
portland
    media
    100 at 000031|000032|000034|000035|000037|000038
    endmedia
#
sacramento
    media
    100 at 000131|000132|000134|000135|000137|000138
    endmedia
```

As CODE EXAMPLE 2-6 shows, a server configuration file consists of multiline entries for each client. A pound character (#) indicates a comment line. Anything to the right of a comment line is ignored.

CODE EXAMPLE 2-7 shows the format for a Sun SAM-Remote server configuration file.

CODE EXAMPLE 2-7 Server Configuration File Format

```
client_name
    [ parameter1 ]
    media
        eq media_type regex
        [ eq media_type regex ]
        [ . . . ]
    endmedia
```

The following steps show how to write the server configuration file.

a. Write the *client_name* field.

The *client_name* defines the network name for each client to be served by this invocation of the Sun SAM-Remote daemon. The first character in the *client_name* must be the first character in the line. The *client_name* can be specified as either the network name, an IP address, or a fully qualified domain name.

The *parameter* (if specified) and media specifications following a *client_name*, and up to the next client definition, are specific to this client. The *parameter* and *media* definitions must be indented with white space or tab characters.

b. (Optional) Write the *parameter* field.

The parameter line is expressed in a *keyword = value* pair. You can use the *parameter* field to specify the network block size. The `net_block_size` parameter specifies the network block size to be used by this client's socket, in kilobytes. The format for this parameter is as follows:

```
net_blk_size=size
```

For *size*, specify an integer from $4 \leq size \leq 64$. The default is 4, which specifies 4096 bytes.

c. Write the `media` and `endmedia` keyword fields.

The `media` and `endmedia` keywords are required in the server configuration file. They define the media archive volumes that a client can use. These media associations are specified as follows:

CODE EXAMPLE 2-8 The Media Specification in the Server Configuration File

```
media
    eq media_type regex
    [ eq media_type regex ]
    [ . . . ]
endmedia
```

The `media` and `endmedia` keywords delimit the media definition area of the Sun SAM-Remote server configuration file. The `eq media_type regex` lines are the media definition lines. The elements of the media type specification are as follows:

Argument	Definition
<code>eq</code>	The equipment ordinal of a library. Network-attached libraries with mixed media can have more than one <code>eq media_type regex</code> line, so specify a different <code>eq media_type regex</code> line for each media type.
<code>media_type</code>	The two-character specific media type. Note that the generic media type specifications that are valid in the <code>mcf</code> file are not valid for the <code>media_type</code> specification. The specification must be for a specific media type (<code>lt</code> , for example). For information about valid media types, see the <code>mcf(4)</code> man page. Specify more than one media definition line if you have a network-attached library with more than one media type.
<code>regex</code>	The volume serial names (VSNs) of the cartridges to which the files will be archived. Each VSN specified must be expressed as an extended regular expression. For information about extended regular expressions, see the <code>egrep(1)</code> man page. You can specify more than one media definition line for each <code>media_type</code> , which gives you flexibility in defining media. For example, the following is a valid media type definition: <pre>media 100 lt VSN1 100 lt VSN2 endmedia</pre> For information about regular expressions, see the <code>regcomp(3C)</code> man page.

Note – Do not allow the same physical media cartridges to be used by more than one client. In addition, if the Sun SAM-Remote server has its own file system outside of the Sun SAM-Remote environment, it is not recommended that a cartridge be used by both the client and the server.

▼ To Enable Archiving

The following steps enable archiving and complete the configuration process.

1. **Verify the `archiver.cmd` file on the client.**

Depending on your configuration, you might need to perform the following tasks:

- Make sure that the VSNs defined in the server configuration file are assigned to the correct archive sets in the `archiver.cmd` file.
- Remove the following directives from the `archiver.cmd` file on the Sun SAM-Remote client if these directives apply to archive sets to be archived to the library connected to the Sun SAM-Remote server:
 - `-tapenonstop`
 - `-offline_copy direct`

2. Issue the `samd(1M)` command with its `start` option to start the Sun SAM-FS or Sun SAM-QFS processes on the server and on the client(s).

To ensure that the new configuration files on the server and clients are read, you must start or restart your Sun SAM-FS or Sun SAM-QFS software.

Enter the following command on the clients and the server:

```
server# samd start
```

For more complete instructions about starting and restarting Sun SAM-FS and Sun SAM-QFS, see the *Sun QFS, Sun SAM-FS, and Sun SAM-QFS Installation and Configuration Guide*.

3. Invoke `samu(1M)` on the server and the client(s).

The goal of this step is to verify the connection between hosts. Use the `samu(1M)` utility's `s` and `R` displays to show the status of Sun SAM-Remote connections. For more information on `samu(1M)`, see the `samu(1M)` man page or see the *Sun SAM-FS and Sun SAM-QFS Storage and Archive Management Guide*.

CODE EXAMPLE 2-9 shows the `samu(1M)` status `s` display on the Sun SAM-Remote client, `portland`. Note the device type `sc`, which represents the Sun SAM-Remote client. The message below that line indicates that a connection with the server `chicago` has been established.

CODE EXAMPLE 2-9 Client `samu(1M)` `s` Display

```
Device status          samu  4.0.5 Wed May 02 14:44:44
License: License never expires.

ty      eq state  device_name          fs status  pos
ms      1 on     samfs1               1 m-----

md      10 on     /dev/dsk/c1t1d0s0    1  -----

md      12 on     /dev/dsk/c1t2d0s0    1  -----
```

CODE EXAMPLE 2-9 Client samu(1M) s Display (Continued)

s9	35	on	/dev/samst/c0t5u0	35	m-----r
			move complete		
lt	36	on	/dev/rmt/0cbn	35	-----p
			empty		
lt	37	on	/dev/rmt/1cbn	35	-----p
			empty		
lt	38	on	/dev/rmt/2cbn	35	--l-----r
			idle		
lt	39	on	/dev/rmt/3cbn	35	--l-----r
			idle		
sc	200	on	/etc/opt/SUNWsamfs/rmt200	200	-----r
			server chicago connected		
rd	201	on	/dev/samrd/rd0	200	-----r
rd	202	on	/dev/samrd/rd1	200	-----r
hy	203	on	historian	203	-----

CODE EXAMPLE 2-10 shows the samu(1M) status s display on the Sun SAM-Remote server chicago. Note the device type ss, which represents the Sun SAM-Remote server. This display indicates that this system is a Sun SAM-Remote server.

CODE EXAMPLE 2-10 Server samu(1M) s Display on chicago

Device status						samu	4.0.5	Tue Apr 24 14:49:43
License: License never expires.								
ty	eq	state	device_name	fs	status	pos		
ms	1	on	samfs1	1	m-----			
md	11	on	/dev/dsk/c2t6d0s0	1	-----			
md	12	on	/dev/dsk/c2t6d0s1	1	-----			
ss	50	on	/etc/opt/SUNWsamfs/rmt200	50	-----r			
sl	100	on	/dev/samst/c0t3u0	100	m-----r			
at	101	on	/dev/rmt/0cbn	100	-----p			
			initializing					
at	102	on	/dev/rmt/1cbn	100	-----p			
			initializing					
hy	103	on	historian	103	-----			

CODE EXAMPLE 2-11 shows the `samu(1M)` Sun SAM-Remote `R` display from the Sun SAM-Remote server `chicago`.

CODE EXAMPLE 2-11 Server `samu(1M)` `R` Display on `chicago`

```
Remote server eq: 50                addr: 00001ca0 4.0.5 Wed May 02
14:55:37
License: License never expires.

message:

Client: portland                    cache action - bypass cache
cache size - 0                      client index - 0
cache left - 0                      network block size - 4096
max file size - 0                   flags - c0000000
min file size - 8                   no-cache connected
```

Note – The cache information in CODE EXAMPLE 2-11 is not useful, and it will be removed from the `samu(1M)` `R` display in the Sun SAM-Remote 4.1 release.

If you have multiple Sun SAM-Remote clients, you can scroll through the clients by pressing the `CONTROL-f` key sequence.

In CODE EXAMPLE 2-11, the connected client is named `portland`. The `client index` field indicates that this client is the zero of a possible 0 through 9 clients defined for this server daemon. The maximum file size, minimum file size, and network block size are listed in bytes. Flags indicate the state of the connection, as follows:

TABLE 2-1 The `samu(1M)` `R` Display Flags

Flag	Meaning
0x00000000	No connection.
0xc0000000	A connection has been established.

4. From the server, use the `samu(1M)` utility to ensure that the catalog is available on the client(s).

For each client, you should be able to view the Sun SAM-Remote catalog that is available for that client by using the `samu(1M)` utility's `v` display to show VSNs. From `samu(1M)`, enter the following:

```
:v eq
```

The *eq* must be the equipment ordinal of the Sun SAM-Remote client daemon as defined in the *mcf* file.

CODE EXAMPLE 2-12 shows a *samu(1M)* display from *chicago*. This display was obtained by specifying *:v 200* on *chicago*. It shows the volumes that *portland* can access from *chicago*.

CODE EXAMPLE 2-12 Volumes Available as Viewed From *chicago*

```
Robot VSN catalog by slot : eq 200 samu 4.0.5 Wed May 02 15:24:13
License: License never expires.                                count 32
slot      access      time  count use flags      ty vsn
  1      2003/01/02  10:40   0   0% -il-o-b-R-U-   at 000032
  2      2003/01/02  11:41   0   0% -il-o-b-R---   at 000034
  3      2003/01/02  12:42  170  91% -il-o-b----- at 000035
  4      2003/01/02  13:43   20   7% -il-o-b----- at 000037
  5      2003/01/02  14:44   0   0% -il-o-b----- at 000038
  6      2003/01/02  13:41   0   0% -il-o-b----- at 000031
```

5. From the client(s), issue the *archiver(1M)* command and its *-A* option.

In this step, you verify that archiving is taking place from the client to the server. You can do this by using the *archiver(1M)* command and its *-A* option. This option enables a listing to be written from the archiver, and this listing includes the VSNs from the server. For information about this command, see the *archiver(1M)* man page.

If files are not archiving, see the *Sun SAM-FS and Sun SAM-QFS Storage and Archive Management Guide* For information about how to troubleshoot the archiver.

Recycling With Sun SAM-Remote

This chapter contains information about recycling with Sun SAM-Remote. Sun Microsystems recommends recycling in a Sun SAM-Remote environment only under the very specific circumstances described in this chapter. The restrictions on recycling are described in this chapter. They must be followed exactly, or data loss can result. It is important that you follow the recommendations in this chapter because there is no enforcement of these restrictions in the Sun SAM-FS or Sun SAM-QFS software.

Because the recycling process involves freeing space on cartridges for more data, it is possible for the recycler to destroy important data on archive cartridges if the recycling process is not configured properly.



Caution – Using the recycler in a Sun SAM-Remote environment requires a complete understanding of each step of the recycler. Executing commands in the wrong order, or on the wrong system, can result in an irreversible loss of data. Make sure you have analyzed a command's actions before executing any command, such as `tplabel(1M)`, that can delete data on the Sun SAM-Remote client or the Sun SAM-Remote server.

It is very important that recycling activities on the Sun SAM-Remote server and the Sun SAM-Remote client not overlap. The result could be accidental relabeling of cartridges and irreversible loss of data.

You cannot recycle cartridges that contain removable media files.

In a Sun SAM-Remote client and server environment, the client and server are unaware of each other's file systems, data files, and inode files. The server and the client each must have exclusive use of a certain set of cartridges. Each must never use the other's cartridges. You can prevent accidental recycling of VSNs used by Sun SAM-Remote clients by creating a `no_recycle` list in the Sun SAM-Remote server's `/etc/opt/SUNWsamfs/recycler.cmd` file. Be careful of using the `chmed(1M)` command's `+c` option on volumes in a `no_recycle` list, however. When you use this command to set the recycling flag (`+c`) on a volume, that action overrides the

`no_recycle` list in the `/etc/opt/SUNWsamfs/recycler.cmd` file.

Do not attempt to recycle volumes on the Sun SAM-Remote server and Sun SAM-Remote client on the same day.

Recycling in a Sun SAM-Remote environment should occur only if the following conditions are present:

- Each VSN in the system is used by one client system or by the server. There cannot be files from multiple systems on any VSN.
- No Sun SAM-Remote client has catalog entries for any VSNs other than those VSNs containing that client's archive images. The *regex* in the server configuration file's media definition lines (the *eq media_type regex* lines) must agree with the volumes specified in the client catalog. In addition, the *regex* specifications in the client catalogs cannot specify the same volumes.
- The archiving is performed on an archive set basis. When using Sun SAM-Remote, recycling must be performed by archive set, not by library.

This chapter describes two methods for enabling recycling using a Sun SAM-Remote client and server. The methods are as follows:

- “Recycling in a Sun SAM-Remote Environment—Method 1” on page 24
- “Recycling in a Sun SAM-Remote Environment—Method 2” on page 49

Recycling in a Sun SAM-Remote Environment—Method 1

The procedures in this section describe one method for enabling recycling in a Sun SAM-Remote environment. Throughout this section, the example environment is one in which the server is named `sky` and the client is named `zeke`. This procedure shows how to configure Sun SAM-Remote to create archive copies of files on cartridges in two different libraries. Archive copy 1 will be written using a StorageTek library that is local to `zeke`. Archive copy 2 will be written remotely, using an ADIC library attached to `sky`. Pertinent files for these two systems are shown in the following sections.



Caution – Use the the recycler in a Sun SAM-Remote environment only after following the steps in this procedure completely and only after testing your configuration to see that correct recycling is taking place.

Configuration Files for Server sky

The server must have Sun SAM-Remote configuration information in its `mcf` file and in its server configuration file. The following code examples show these files.

CODE EXAMPLE 3-1 shows the `mcf` file on server `sky`.

CODE EXAMPLE 3-1 The `mcf` File on Server `sky`

```
# This is the mcf file for the server (sky).
# The server parameters file (rmt1000) points
#   back to the correct automated library's equipment number
#   (70) for the ADIC Scalar 1000.
#
samfs1          100   ma   samfs1   on
/dev/dsk/c0t0d0s5 110   mm   samfs1   on   /dev/rdisk/c0t0d0s5
/dev/dsk/c3t2d0s3 120   mr   samfs1   on   /dev/rdisk/c3t2d0s3
/dev/dsk/c3t2d0s4 121   mr   samfs1   on   /dev/rdisk/c3t2d0s4

samfs2          139   ma   samfs2   on
/dev/dsk/c3t4d0s3 140   mm   samfs2   on   /dev/rdisk/c3t4d0s3
/dev/dsk/c3t4d0s4 141   mr   samfs2   on   /dev/rdisk/c3t4d0s4

# ADIC Scalar 1000
/dev/samst/c0t0u0 70 rb adic1 - /var/opt/SUNWsamfs/catalog/adic1
/dev/rmt/0bn      71   at   adic1   on
/dev/rmt/1bn      72   at   adic1   on
/dev/rmt/2bn      73   at   adic1   on
/dev/rmt/3bn      74   at   adic1   on
/dev/rmt/4bn      75   at   adic1   on
/dev/rmt/5bn      76   at   adic1   on
/dev/rmt/11bn     77   at   adic1   on
/dev/rmt/10bn     78   at   adic1   on
/dev/rmt/9bn      79   at   adic1   on
/dev/rmt/8bn      80   at   adic1   on
/dev/rmt/7bn      81   at   adic1   on
/dev/rmt/6bn      82   at   adic1   on

# Define Sun SAM-Remote server skyrs
/etc/opt/SUNWsamfs/rmt1000 1000 ss skyrs on
```

CODE EXAMPLE 3-2 shows the server configuration file on server sky.

CODE EXAMPLE 3-2 The Server Configuration File on Server sky

```
# Server configuration file /etc/opt/SUNWsamfs/rmt1000 on sky.
# The eq of the automated library MUST match the eq of the
# automated library that you want to use in the mcf file.

zeke
  media
    70 at 00002[0-9]
  endmedia
```

Configuration Files for Client zeke

The client must have Sun SAM-Remote configuration information in its mcf file and in its client configuration file. The following code examples show these files.

CODE EXAMPLE 3-3 shows the mcf file on client zeke.

CODE EXAMPLE 3-3 The mcf File on Client zeke

```
# mcf file for client (zeke)
#
samfs1          10  ms  samfs1  on
/dev/dsk/clt3d0s0  11  md  samfs1  on  /dev/rdisk/clt3d0s0
/dev/dsk/clt3d0s1  12  md  samfs1  on  /dev/rdisk/clt3d0s1
/dev/dsk/clt3d0s3  13  md  samfs1  on  /dev/rdisk/clt3d0s3

# Define a StorageTek L20 with 1 drive and 20 slots (including cap)
/dev/samst/c0t2u0  50  rb  stk_l20  on /var/opt/SUNWsamfs/catalog/L20_cat
/dev/rmt/0hbn     51  lt  stk_l20  on

# Define zeke as a Sun SAM-Remote client using sky as the server
/etc/opt/SUNWsamfs/sky 200  sc  skyrs  on /var/opt/SUNWsamfs/catalog/sky_cat
/dev/samrd/rd0       201  rd  skyrs  on
/dev/samrd/rd1       202  rd  skyrs  on
/dev/samrd/rd2       203  rd  skyrs  on
/dev/samrd/rd3       204  rd  skyrs  on
```

CODE EXAMPLE 3-4 shows the client configuration file on client zeke.

CODE EXAMPLE 3-4 The Client Configuration File on Client zeke

```
# cat /etc/opt/SUNWsamfs/sky
# File /etc/opt/SUNWsamfs/sky on Sun SAM-Remote client zeke:
sky
```

▼ To Configure Recycling—Method 1

The following procedure shows how to configure the recycling process. This procedure includes a test for archiving and recycling. Because of the testing period, this procedure can take a day or two to complete, depending on how frequently files are archived and recycled.

Note – Do not use the `chmed(1M)` command on the server to set the recycling flag (+c) for a client VSN. That action overrides the `no_recycle` list in the `/etc/opt/SUNWsamfs/recycler.cmd` file on the server.

1. Read about the recycler in the *Sun SAM-FS and Sun SAM-QFS Storage and Archive Management Guide*.

Using the recycler in a Sun SAM-Remote environment requires a complete understanding of the steps in the recycling process. If you have not already familiarized yourself with the recycling process, take time now.

2. Make sure that the Sun SAM-Remote client and server are configured properly and that archiving is occurring.

For more information on configuring and verifying your Sun SAM-Remote environment, see “Configuring Sun SAM-Remote Software” on page 7, which contains detailed information about configuring the Sun SAM-Remote client and server. That procedure includes steps for ensuring that archiving is taking place.

3. Edit the `archiver.cmd` file on the client system and add recycling directives.

In this example, the recycling is performed by archive set, not by library. The directives specifying that recycling be done by archive set must appear in the `archiver.cmd` file.

CODE EXAMPLE 3-5 shows the `archiver.cmd` file on client `zeke`. This file has been edited to communicate with the recycler.

CODE EXAMPLE 3-5 The `archiver.cmd` File on Client `zeke`

```
# This is file /etc/opt/SUNWsamfs/archiver.cmd
# on Sun SAM-Remote client zeke.
#
# wait

logfile = /var/opt/SUNWsamfs/archiver/archiver.log
trace = /var/opt/SUNWsamfs/trace/archiver all

interval = 1m

no_archive tmp
no_archive .

archmax = 1t 2G
archmax = at 5G

drives = skyr4 # use up to four drives for remote archiving.

fs = samfs1
  1 4h
archiveset testdir0
  1 1m
  2 1m
defaultset .
  1 1m
  2 1m

params

# Start with mingain high to reduce workload.
# If you need more recycling, reduce mingain.
# If too much recycling, increase High Water Mark.
archiveset.1 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
archiveset.1 -recycle_ignore
defaultset.1 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
defaultset.1 -recycle_ignore

# Remote directives.
# Use up to three drives per archive set.
# Load will split to two drives at 100m, to three drives at 150m.
archiveset.2 -drives 3 -drivemin 50m
defaultset.2 -drives 3 -drivemin 50m
```

CODE EXAMPLE 3-5 The archiver.cmd File on Client zeke (Continued)

```
# Remote directives.
# Start with mingain high to reduce workload.
# If you need more recycling, reduce mingain.
# If too much recycling, increase High Water Mark.
archiveset.2 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
archiveset.2 -recycle_ignore
defaultset.2 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
defaultset.2 -recycle_ignore
endparams

vsns
samfs1.1      lt 000173      # local copy.
archiveset.1  lt ^CEL       # local copy.
archiveset.2  at 00002[0-4] # remote copy, sky ait-2
                                     # tapes 20 through 24.
defaultset.1  lt ^CSM       # local copy.
defaultset.2  at 00002[5-9] # remote copy, sky ait-2
                                     # tapes 25 through 29.
endvsns
```

The directives shown in CODE EXAMPLE 3-5 perform as follows:

- The `-recycle_hwm` directive sets the library's high-water mark for the archive set. When the utilization of the VSNs exceeds this percentage, recycling of the archive set begins.
 - The `-recycle_ignore` directive is inserted only temporarily. This directive prevents recycling from occurring until you have configured and tested your environment. You can remove this directive in a later step.
 - The `-recycle_mingain` directive is set high to limit the amount of work needed to regain space. That is, this directive is set high to ensure efficiency.
 - The `-recycle_vsncount 1` directive prevents recycling from overwhelming the system. This directive specifies that the recycler drain one VSN at a time. When the first VSN is drained, a second is selected to begin draining. So at any moment in time, there is one VSN in the queue to be relabeled and one VSN in the queue to be drained.
- 4. Edit the `recycler.cmd` file on the client and specify a log file to receive recycling log output.**

The following `recycler.cmd` file on client `zeke` has been edited to specify a recycler log file:

CODE EXAMPLE 3-6 The `recycler.cmd` File on Client `zeke`

```
#
# This is the /etc/opt/SUNWsamfs/recycler.cmd file
# on client zeke.
#
logfile = /var/opt/SUNWsamfs/log/recycler
```

5. Verify that the `archiver.cmd` file on the server is written to specify recycling by archive set.

When using Sun SAM-Remote, you must specify that recycling be performed on an archive set basis, not by library. The directives specifying that recycling be done by archive set must appear in the `archiver.cmd` file.

CODE EXAMPLE 3-7 shows the `archiver.cmd` file on server `sky`. This file specifies archiving by archive set.

CODE EXAMPLE 3-7 The `archiver.cmd` File on Server `sky`

```
# This is the archiver.cmd for the server (sky).
#
# Number of drives: 10
# Number of Mounted Filesystems: 1
# Number of Tests per Filesystem: 1
# Number of Archive Copies per Test: 2

#wait
#trace = /var/opt/SUNWsamfs/trace/archiver all

logfile = /var/opt/SUNWsamfs/log/archiver
interval = 1m
no_archive .
archmax = at 5G
drives = adicl 6

fs = samfs1
    1 4h
testset testdir0
    1 1m
    2 1m
allsam1 .
    1 1m
```


CODE EXAMPLE 3-7 The archiver.cmd File on Server sky (Continued)

```
2 1m

params
allsaml.1 -drives 4 -drivemin 50m
allsaml.1 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
allsaml.1 -recycle_ignore
allsaml.2 -drives 4 -drivemin 50m
allsaml.2 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
allsaml.2 -recycle_ignore
testset.1 -drives 4 -drivemin 50m
testset.1 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
testset.1 -recycle_ignore
testset.2 -drives 4 -drivemin 50m
testset.2 -recycle_hwm 60 -recycle_mingain 90 -recycle_vsncount 1
testset.2 -recycle_ignore
endparams

vsns
samfs1.1 at 000000
allsaml.1 at 00000[1-5] # vsns 1 through 5.
allsaml.2 at 00000[6-9] # vsns 6 through 9.
testset.1 at 00001[0,4] # vsns 10 and 14.
testset.2 at 00001[5,9] # vsns 15 and 19.
endvsns
```

6. Edit the recycler.cmd file on the server.

Use an editor to modify the file to specify the following items:

- A recycler log file to receive output from the recycler.
- A `no_recycle` directive for the Sun SAM-Remote client's VSNs. The Sun SAM-Remote client is configured to write its copy 2 archive copies to cartridges in the Sun SAM-Remote server's library. The `no_recycle` directive is necessary to prevent the VSNs being used by the Sun SAM-Remote client for archiving from being recycled by the Sun SAM-Remote server.

The following `recycler.cmd` file on server sky has been edited to specify a recycler log file:

CODE EXAMPLE 3-8 The recycler.cmd File on Server sky

```
#
# This is the /etc/opt/SUNWsamfs/recycler.cmd file
# on Sun SAM-Remote server sky.
#
```

CODE EXAMPLE 3-8 The `recycler.cmd` File on Server `sky`

```
logfile = /var/opt/SUNWsamfs/recycler/recycler.log
adicl -ignore
no_recycle at 00002[0-9] # Prevents VSNs assigned to zeke from
                        # being recycled.
```

7. Use the `sam-recycler(1M)` command to test the recycler on the Sun SAM-Remote client.

Run the recycler on the Sun SAM-Remote client system. This is a test to see if the recycler properly acknowledges the devices and VSNs specified in the configuration files. This testing is important because if the recycler detects that the system it is running on has no archive images on a particular VSN listed in any of that system's catalogs (including the historian catalog), the `recycler.sh` script can call for the cartridge to be labeled. Labeling a cartridge destroys all data on the cartridge. There is no communication between the Sun SAM-Remote client and the Sun SAM-FS or Sun SAM-QFS servers to inform each side of the presence of archive copies. All such information is provided locally from local Sun SAM-FS or Sun SAM-QFS file systems.

For example, you can use the following command to perform the initial test of the recycler:

```
zeke# sam-recycler -dvx
```

The recycler runs and logs its activity to the recycler log file. The recycler log file is defined in the `recycler.cmd` file. For more information about the `sam-recycler(1M)` command, see the `sam-recycler(1M)` man page.

8. Examine the recycler log file.

You are looking for the following message:

```
Recycling is ignored on this archive set.
```

CODE EXAMPLE 3-9 shows a sample log file.

CODE EXAMPLE 3-9 Recycler Log File on Client `zeke`

```
# recycler.log from client zeke.

===== Recycler begins at Mon Jun  4 09:49:41 2001 =====
Initial 7 catalogs:

0  Family: stk_l20                Path: /var/opt/SUNWsamfs/catalog/L20_cat
```

CODE EXAMPLE 3-9 Recycler Log File on Client zeke (Continued)

```

Vendor: STK                      Product: L20
SLOT          ty      capacity      space vsn
  0           lt      33.0G          33.0G 000173
  1           lt      32.8G          44.1M CEL170
  2           lt      33.0G          33.0G CEL139
  4           lt      32.8G          16.8G CFC504
  5           lt      33.0G          33.0G CFC503
  6           lt      32.9G           0    CSM689
  7           lt      32.9G          19.6G CSM690
  8           lt      33.0G          33.0G CSM691
  9           lt      33.0G          33.0G CSM692
 10           lt      10.0G          10.0G CLN018
 11           lt      33.0G          33.0G 000766
Total Capacity: 339.2G bytes, Total Space Available: 244.3G bytes
Volume utilization 27%, high 95% VSN_min 50%
Recycling is ignored on this robot.

1 Family: skyrs                  Path: /var/opt/SUNWsamfs/catalog/sky_cat
Vendor: (NULL)                  Product: (NULL)
SLOT          ty      capacity      space vsn
  0           at      48.5G          23.3G 000020
  1           at      23.8G          23.8G 000021
  2           at      48.5G          48.5G 000022
  3           at      48.5G          48.5G 000023
  4           at      48.5G          48.5G 000024
  5           at      48.5G           2.6G 000025
  6           at      48.5G          361.4k 000026
  7           at      48.5G          48.5G 000027
  8           at      48.5G          48.5G 000028
  9           at      48.5G           0    000029
Total Capacity: 460.8G bytes, Total Space Available: 292.5G bytes
Volume utilization 36%, high 95% VSN_min 50%
Recycling is ignored on this robot.

2 Family: hy                     Path: /var/opt/SUNWsamfs/catalog/historian
Vendor: Sun SAM-FS              Product: Historian
SLOT          ty      capacity      space vsn
(no VSNs in this media changer)
Total Capacity: 0 bytes, Total Space Available: 0 bytes
Volume utilization 0%, high 95% VSN_min 50%
Recycling is ignored on this robot.

```

CODE EXAMPLE 3-9 Recycler Log File on Client zeke (Continued)

```
3 Family: defaultset.1          Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS              Product: Archive set
SLOT                             ty    capacity      space vsn
  0                             lt    33.0G         33.0G 000766
  1                             lt    33.0G         33.0G 000173
  2                             lt    32.9G         0      CSM689
  3                             lt    32.9G         19.6G CSM690
  4                             lt    33.0G         33.0G CSM691
  5                             lt    33.0G         33.0G CSM692
Total Capacity: 197.6G bytes, Total Space Available: 151.5G bytes
Volume utilization 23%, high 60% VSN_min 90%
Recycling is ignored on this archive set.
```

```
4 Family: defaultset.2          Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS              Product: Archive set
SLOT                             ty    capacity      space vsn
  0                             lt    32.9G         0      CSM689
  1                             at    48.5G         23.3G 000020
  2                             at    23.8G         23.8G 000021
  3                             at    48.5G         2.6G  000025
  4                             at    48.5G         361.4k 000026
  5                             at    48.5G         48.5G 000027
  6                             at    48.5G         48.5G 000028
  7                             at    48.5G         0      000029
Total Capacity: 348.0G bytes, Total Space Available: 146.8G bytes
Volume utilization 57%, high 60% VSN_min 90%
Recycling is ignored on this archive set.
```

```
5 Family: archiveset.1          Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS              Product: Archive set
SLOT                             ty    capacity      space vsn
  0                             lt    32.8G         44.1M CEL170
  1                             lt    32.8G         16.8G CFC504
  2                             lt    33.0G         33.0G CFC503
Total Capacity: 98.6G bytes, Total Space Available: 49.8G bytes
Volume utilization 49%, high 60% VSN_min 90%
Recycling is ignored on this archive set.
```

CODE EXAMPLE 3-9 Recycler Log File on Client zeke (Continued)

```

6 Family: archiveset.2          Path: /etc/opt/SUNWsamfs/archiver.cmd
Vendor: Sun SAM-FS              Product: Archive set
SLOT                            ty      capacity      space vsn
  0                             at      48.5G        23.3G 000020
  1                             at      23.8G        23.8G 000021
  2                             at      48.5G        48.5G 000022
  3                             at      48.5G        48.5G 000023
  4                             at      48.5G        48.5G 000024
Total Capacity: 218.0G bytes, Total Space Available: 192.8G bytes
Volume utilization 11%, high 60% VSN_min 90%
Recycling is ignored on this archive set.

```

21 VSNs:

```

---Archives---      -----Percent-----      defaultset.1
-----Status-----      Count      Bytes      Use Obsolete Free      Library:Type:VSN
in multiple sets      0          0          0   100      0      stk_l20:lt:CSM689
partially full      111        2.8G      8    31      61      stk_l20:lt:CSM690
empty VSN            0          0          0    0      100     stk_l20:lt:000173
empty VSN            0          0          0    0      100     stk_l20:lt:CSM691
empty VSN            0          0          0    0      100     stk_l20:lt:CSM692
empty VSN            0          0          0    0      100     stk_l20:lt:000766

```

```

---Archives---      -----Percent-----      defaultset.2
-----Status-----      Count      Bytes      Use Obsolete Free      Library:Type:VSN
no-data VSN          0          0          0   100      0      skyrs:at:000029
no-data VSN          0          0          0   99      1      skyrs:at:000026
partially full      111        2.8G      6    88      6      skyrs:at:000025
empty VSN            0          0          0    0      100     skyrs:at:000028
empty VSN            0          0          0    0      100     skyrs:at:000027

```

```

---Archives---      -----Percent-----      archiveset.1
-----Status-----      Count      Bytes      Use Obsolete Free      Library:Type:VSN
no-data VSN          0          0          0   99      1      stk_l20:lt:CEL170
partially full      677        2.3G      8    40      52     stk_l20:lt:CFC504
empty VSN            0          0          0    0      100     stk_l20:lt:CFC503

```

```

---Archives---      -----Percent-----      archiveset.2
-----Status-----      Count      Bytes      Use Obsolete Free      Library:Type:VSN
in multiple sets      0          0          0   51      49     skyrs:at:000020
empty VSN            0          0          0    0      100     skyrs:at:000022

```

CODE EXAMPLE 3-9 Recycler Log File on Client zeke (Continued)

```
empty VSN          0      0      0      0      100  skyrs:at:000023
empty VSN          0      0      0      0      100  skyrs:at:000024
in multiple sets   0      0      0      0      100  skyrs:at:000021

-----Status-----  ---Archives---  -----Percent-----  stk_l20
Count      Bytes      Use Obsolete Free  Library:Type:VSN
empty VSN   0      0      0      0      100  stk_l20:lt:CLN018
partially full  13    80.3k  0      0      100  stk_l20:lt:CEL139

Recycler finished.

===== Recycler ends at Mon Jun  4 09:49:53 2001 =====
```

9. Issue the `sam-recycler(1M)` command from the Sun SAM-Remote server to test the recycler.

Make sure that the recycler is not recycling any VSNs reserved for the Sun SAM-Remote client.

For example:

```
zeke# sam-recycler -dvx
```

The preceding command runs the recycler and writes its activity to the recycler log file. For more information about the `sam-recycler(1M)` command, see the `sam-recycler(1M)` man page.

CODE EXAMPLE 3-10 shows a sample recycler log file.

CODE EXAMPLE 3-10 The Recycler Log File

```
# recycler.log file from server sky.

===== Recycler begins at Mon Jun  4 09:50:44 2001 =====
Initial 6 catalogs:

0  Family: adicl          Path: /var/opt/SUNWsamfs/catalog/adicl
   Vendor: ADIC          Product: Scalar 1000
   SLOT      ty      capacity      space vsn
   0          at      1.3G          1.2G 000001
   1          at      1.3G          1.3G 000002
   2          at      1.3G          1.3G 000004
   3          at      48.5G         0    000010
   4          at      48.5G         0    000011
```

CODE EXAMPLE 3-10 The Recycler Log File (Continued)

```
5          at          48.5G          43.5G 000018
6          at          48.5G          0    000019
7          at          48.5G          23.3G 000020
8          at          23.8G          23.8G 000021
9          at          48.5G          48.5G 000022
10         at          48.5G          48.5G 000023
11         at          48.5G          48.5G 000024
12         at          48.5G          2.6G 000025
13         at          48.5G          361.4k 000026
14         at          48.5G          48.5G 000027
15         at          48.5G          48.5G 000028
16         at          48.5G          0    000029
17         at          1.3G          1.3G 000005
18         at          48.5G          48.5G 000016
19         at          23.8G          23.8G CLN001
20         at          23.8G          23.8G CLN002
21         at          23.8G          23.8G CLN004
22         at          23.8G          23.8G CLN003
23         at          48.5G          421.6M 000015
24         at          1.3G          1.3G 000000
25         at          48.5G          0    000013
26         at          1.3G          1.3G 000003
27         at          48.5G          43.6G 000007
28         at          48.5G          41.8G 000008
29         at          48.5G          46.9G 000006
30         at          48.5G          48.3G 000009
31         at          48.5G          0    000014
32         at          48.5G          0    000012
33         at          48.5G          40.1G 000017
Total Capacity: 1.2T bytes, Total Space Available: 708.7G bytes
Volume utilization 43%, high 95% VSN_min 50%
Recycling is ignored on this robot.

1 Family: hy          Path: /var/opt/SUNWsamfs/catalog/historian
Vendor: Sun SAM-FS          Product: Historian
SLOT          ty          capacity          space vsn
(no VSNs in this media changer)
Total Capacity: 0 bytes, Total Space Available: 0 bytes
Volume utilization 0%, high 95% VSN_min 50%
Recycling is ignored on this robot.
```

CODE EXAMPLE 3-10 The Recycler Log File (Continued)

```
2 Family: testset.1          Path: /etc/opt/SUNWsamfs/archiver.cmd
  Vendor: Sun SAM-FS        Product: Archive set
  SLOT      ty      capacity      space vsn
    0       at      48.5G          0 000010
    1       at      48.5G          0 000014
  Total Capacity: 97.1G bytes, Total Space Available: 0 bytes
  Volume utilization 100%, high 60% VSN_min 90%: *** Needs recycling ***
  Recycling is ignored on this archive set.

3 Family: testset.2          Path: /etc/opt/SUNWsamfs/archiver.cmd
  Vendor: Sun SAM-FS        Product: Archive set
  SLOT      ty      capacity      space vsn
    0       at      48.5G          0 000019
    1       at      48.5G          421.6M 000015
  Total Capacity: 97.1G bytes, Total Space Available: 421.6M bytes
  Volume utilization 99%, high 60% VSN_min 90%: *** Needs recycling ***
  Recycling is ignored on this archive set.

4 Family: allsam1.1          Path: /etc/opt/SUNWsamfs/archiver.cmd
  Vendor: Sun SAM-FS        Product: Archive set
  SLOT      ty      capacity      space vsn
    0       at      1.3G           1.2G 000001
    1       at      1.3G           1.3G 000002
    2       at      1.3G           1.3G 000004
    3       at      1.3G           1.3G 000005
    4       at      1.3G           1.3G 000003
  Total Capacity: 6.5G bytes, Total Space Available: 6.3G bytes
  Volume utilization 3%, high 60% VSN_min 90%
  Recycling is ignored on this archive set.

5 Family: allsam1.2          Path: /etc/opt/SUNWsamfs/archiver.cmd
  Vendor: Sun SAM-FS        Product: Archive set
  SLOT      ty      capacity      space vsn
    0       at      48.5G          43.6G 000007
    1       at      48.5G          41.8G 000008
    2       at      48.5G          46.9G 000006
    3       at      48.5G          48.3G 000009
  Total Capacity: 194.2G bytes, Total Space Available: 180.6G bytes
```


CODE EXAMPLE 3-10 The Recycler Log File (Continued)

```
Volume utilization 6%, high 60% VSN_min 90%  
Recycling is ignored on this archive set.
```

```
Need to select candidate for media changer testset.1 to free up 39.8G bytes.  
Quantity of data to move limited to (no limit) bytes and 1 VSNs.  
Checking 000010. Need to free 39.8G, quantity limit: (no limit), VSN count: 1.  
  VSN is in correct media changer... good.  
  VSN is not already recycling... good.  
  VSN has no request files... good.  
  VSN has no 'archive -n' files...good.  
  VSN was not specified as "no_recycle" in recycler.cmd file... good.  
  VSN does not exceed VSN count limit... good.  
  VSN does not exceed data quantity limit... good.  
  VSN meets minimum gain requirement.  
  Recycling is ignored on this media changer - VSN not marked for recycling.  
Checking 000014. Need to free 0E, quantity limit: (no limit), VSN count: 0.  
  VSN is in correct media changer... good.  
  VSN is not already recycling... good.  
  VSN has no request files... good.  
  VSN has no 'archive -n' files...good.  
  VSN was not specified as "no_recycle" in recycler.cmd file... good.  
  VSN exceeds VSN count limit - skipped.  
Checking 000019. Need to free 0E, quantity limit: (no limit), VSN count: 0.  
  VSN not in correct media changer.  
Checking 000015. Need to free 0E, quantity limit: (no limit), VSN count: 0.  
  VSN not in correct media changer.  
Checking 000001. Need to free 0E, quantity limit: (no limit), VSN count: 0.  
  VSN not in correct media changer.  
Checking 000003. Need to free 0E, quantity limit: (no limit), VSN count: 0.  
  VSN not in correct media changer.  
Checking 000004. Need to free 0E, quantity limit: (no limit), VSN count: 0.  
  VSN not in correct media changer.  
Checking 000005. Need to free 0E, quantity limit: (no limit), VSN count: 0.  
  VSN not in correct media changer.  
Checking 000002. Need to free 0E, quantity limit: (no limit), VSN count: 0.  
  VSN not in correct media changer.  
Checking 000008. Need to free 0E, quantity limit: (no limit), VSN count: 0.  
  VSN not in correct media changer.  
Checking 000007. Need to free 0E, quantity limit: (no limit), VSN count: 0.  
  VSN not in correct media changer.  
Checking 000006. Need to free 0E, quantity limit: (no limit), VSN count: 0.  
  VSN not in correct media changer.  
Checking 000009. Need to free 0E, quantity limit: (no limit), VSN count: 0.  
  VSN not in correct media changer.
```

CODE EXAMPLE 3-10 The Recycler Log File (Continued)

```
Checking 000011. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000029. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000013. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000012. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000026. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000025. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000020. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000017. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000018. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking CLN003. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000021. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000022. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000027. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000028. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000023. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000024. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000016. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking CLN001. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking CLN002. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking CLN004. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
Checking 000000. Need to free 0E, quantity limit: (no limit), VSN count: 0.
  VSN not in correct media changer.
No candidate was found in this media changer.

Need to select candidate for media changer testset.2 to free up 38.8G bytes.
Quantity of data to move limited to (no limit) bytes and 1 VSNs.
Checking 000010. Need to free 38.8G, quantity limit: (no limit), VSN count: 1.
```

CODE EXAMPLE 3-10 The Recycler Log File (Continued)

```
VSN not in correct media changer.
Checking 000014. Need to free 38.8G, quantity limit: (no limit), VSN count: 1.
VSN not in correct media changer.
Checking 000019. Need to free 38.8G, quantity limit: (no limit), VSN count: 1.
VSN is in correct media changer... good.
VSN is not already recycling... good.
VSN has no request files... good.
VSN has no 'archive -n' files...good.
VSN was not specified as "no_recycle" in recycler.cmd file... good.
VSN does not exceed VSN count limit... good.
VSN does not exceed data quantity limit... good.
VSN meets minimum gain requirement.
Recycling is ignored on this media changer - VSN not marked for recycling.
Checking 000015. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN is in correct media changer... good.
VSN is not already recycling... good.
VSN has no request files... good.
VSN has no 'archive -n' files...good.
VSN was not specified as "no_recycle" in recycler.cmd file... good.
VSN exceeds VSN count limit - skipped.
Checking 000001. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000003. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000004. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000005. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000002. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000008. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000007. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000006. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000009. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000011. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000029. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000013. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000012. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000026. Need to free 0E, quantity limit: (no limit), VSN count: 0.
```

CODE EXAMPLE 3-10 The Recycler Log File (Continued)

```

VSN not in correct media changer.
Checking 000025. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000020. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000017. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000018. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN003. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000021. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000022. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000027. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000028. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000023. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000024. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000016. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN001. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN002. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking CLN004. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
Checking 000000. Need to free 0E, quantity limit: (no limit), VSN count: 0.
VSN not in correct media changer.
No candidate was found in this media changer.
34 VSNs:

```

		---Archives---		-----Percent-----			testset.1
-----Status-----	Count	Bytes	Use	Obsolete	Free	Library:Type:VSN	
no-data VSN	0	0	0	100	0	adic1:at:000010	
no-data VSN	0	0	0	100	0	adic1:at:000014	

		---Archives---		-----Percent-----			testset.2
-----Status-----	Count	Bytes	Use	Obsolete	Free	Library:Type:VSN	
no-data VSN	0	0	0	100	0	adic1:at:000019	
partially full	677	2.3G	5	93	2	adic1:at:000015	

		---Archives---		-----Percent-----			allsam1.1
--	--	----------------	--	-------------------	--	--	-----------

CODE EXAMPLE 3-10 The Recycler Log File (Continued)

```

-----Status-----      Count      Bytes      Use  Obsolete  Free      Library:Type:VSN
partially full           97      173.8M      1     9       90      adic1:at:000001
no-data VSN              0         0           0     2       98      adic1:at:000003
no-data VSN              0         0           0     2       98      adic1:at:000004
empty VSN                0         0           0     0      100      adic1:at:000005
empty VSN                0         0           0     0      100      adic1:at:000002

                ---Archives---      -----Percent-----      allsam1.2
-----Status-----      Count      Bytes      Use  Obsolete  Free      Library:Type:VSN
no-data VSN              0         0           0    13       87      adic1:at:000008
partially full           98      1.6G         3     7       90      adic1:at:000007
no-data VSN              0         0           0     3       97      adic1:at:000006
empty VSN                0         0           0     0      100      adic1:at:000009

                ---Archives---      -----Percent-----      adic1
-----Status-----      Count      Bytes      Use  Obsolete  Free      Library:Type:VSN
no-data VSN              0         0           0   100         0      adic1:at:000011
no_recycle VSN           0         0           0   100         0      adic1:at:000029
no-data VSN              0         0           0   100         0      adic1:at:000013
no-data VSN              0         0           0   100         0      adic1:at:000012
no_recycle VSN           0         0           0    99         1      adic1:at:000026
no_recycle VSN           0         0           0    94         6      adic1:at:000025
no_recycle VSN           0         0           0    51        49      adic1:at:000020
no-data VSN              0         0           0    17        83      adic1:at:000017
no-data VSN              0         0           0    10        90      adic1:at:000018
empty VSN                0         0           0     0      100      adic1:at:CLN003
no_recycle VSN           0         0           0     0      100      adic1:at:000021
no_recycle VSN           0         0           0     0      100      adic1:at:000022
no_recycle VSN           0         0           0     0      100      adic1:at:000027
no_recycle VSN           0         0           0     0      100      adic1:at:000028
no_recycle VSN           0         0           0     0      100      adic1:at:000023
no_recycle VSN           0         0           0     0      100      adic1:at:000024
empty VSN                0         0           0     0      100      adic1:at:000016
empty VSN                0         0           0     0      100      adic1:at:CLN001
empty VSN                0         0           0     0      100      adic1:at:CLN002
empty VSN                0         0           0     0      100      adic1:at:CLN004
partially full           12      88.3k         0     0      100      adic1:at:000000

Recycler finished.

===== Recycler ends at Mon Jun  4 09:51:05 2001 =====

```

When selecting VSNs to recycle, examine the last part of the recycler log file that shows columnar data. The leftmost column is the one headed by `Status`. In the preceding recycler log file, the `Status` column indicates several VSNs with a `no_recycle` status. These VSNs are those used by the client.

The best candidates for recycling are those with a 0 value in the `Count`, `Bytes`, and `Use` columns. The last VSN in the list shows its status as `partially full`. This VSN, with `Count`, `Bytes`, and `Use` statistics of 12, 88.3k, and 0, respectively, is not a good candidate for recycling.

10. Analyze the client and server `recycler.log` files.

This step explains how to choose VSNs that are candidates for recycling.

Examine the `recycler.log` file from the client. Toward the end of the file is a `Status` column. VSNs with the following types of status entries are candidates for recycling:

- `no-data VSN`. To recycle a `no-data VSN`, see “To Recycle `no-data VSNs`” on page 44.
- `partially full`. To recycle a `partially full VSN`, see “To Recycle `partially full VSNs`” on page 46.

▼ To Recycle `no-data VSNs`

The `no-data VSNs` are the easiest VSNs to recycle. For these, the `Count`, `Bytes`, and `Use` fields are all 0 (zero).

1. Examine the `recycler.log` file from the client to see if there are any `no-data VSNs`.

Using the example in this chapter, VSNs 000029 and 000026 from the client, `zeke`, can be considered for recycling because they are `no-data VSNs`. This can be determined from CODE EXAMPLE 3-11, which shows the client `recycler.log` file on `zeke`.

CODE EXAMPLE 3-11 The `recycler.log` File on Client `zeke`

```
# From the client zeke recycler.log file:
-----Status-----      ---Archives---      -----Percent-----      defaultset.2
Count      Bytes      Use Obsolete Free      Library:Type:VSN
no-data VSN      0      0      0      100      0      skyrs:at:000029
no-data VSN      0      0      0      99      1      skyrs:at:000026
partially full      111      2.8G      6      88      6      skyrs:at:000025
empty VSN      0      0      0      0      100      skyrs:at:000028
empty VSN      0      0      0      0      100      skyrs:at:000027
```

2. Examine the `recycler.log` file from the server and determine if the VSNs you selected from the previous step are represented identically in the server's recycler log file.

You are trying to affirm that there is no active data from the server archived on those VSNs.

CODE EXAMPLE 3-12 shows the data for the `no_recycle` VSNs in the server's `recycler.log` file. VSNs 000029 and 000026 were selected for recycling from the previous step, and the data in the server's `recycler.log` file is identical to that in the client's `recycler.log` file.

CODE EXAMPLE 3-12 The `recycler.log` File on Server `sky`

```
# From the Server log file:
```

-----Status-----	---Archives---		-----Percent-----			adic1 Library:Type:VSN
	Count	Bytes	Use	Obsolete	Free	
no-data VSN	0	0	0	100	0	adic1:at:000011
no_recycle VSN	0	0	0	100	0	adic1:at:000029zeke
no-data VSN	0	0	0	100	0	adic1:at:000013
no-data VSN	0	0	0	100	0	adic1:at:000012
no_recycle VSN	0	0	0	99	1	adic1:at:000026
no_recycle VSN	0	0	0	94	6	adic1:at:000025
no_recycle VSN	0	0	0	51	49	adic1:at:000020
no-data VSN	0	0	0	17	83	adic1:at:000017
no-data VSN	0	0	0	10	90	adic1:at:000018
empty VSN	0	0	0	0	100	adic1:at:CLN003
no_recycle VSN	0	0	0	0	100	adic1:at:000021
no_recycle VSN	0	0	0	0	100	adic1:at:000022
no_recycle VSN	0	0	0	0	100	adic1:at:000027
no_recycle VSN	0	0	0	0	100	adic1:at:000028
no_recycle VSN	0	0	0	0	100	adic1:at:000023
no_recycle VSN	0	0	0	0	100	adic1:at:000024
empty VSN	0	0	0	0	100	adic1:at:000016
empty VSN	0	0	0	0	100	adic1:at:CLN001
empty VSN	0	0	0	0	100	adic1:at:CLN002
empty VSN	0	0	0	0	100	adic1:at:CLN004
partially full	12	88.3k	0	0	100	adic1:at:000000

3. (Optional) Use the `tplabel(1M)` or `odlabel(1M)` command to relabel the VSN.

If no active data from the server is archived on that VSN, you can relabel the VSN.

Note – This destroys all data on the VSN and reclaims space.

For example, for tape VSN 000029, use the following command:

```
server# tplabel -vsn 000029 -old 000029 at.000029
```

When this VSN 000029 is relabelled, you regain 100 percent of the space on that VSN.

If the media had been a magneto-optical disk, you would have used the `odlabel(1M)` command. For more information on the `odlabel(1M)` command, see the `odlabel(1M)` man page.

4. Devise a recycling schedule.

In a Sun SAM-FS or Sun SAM-QFS environment in which Sun SAM-Remote software is not enabled, you can create a `cron(1)` job so recycling is performed automatically. If Sun SAM-Remote software is enabled, however, do not automate the recycler.



Caution – It is very important that recycling activities not be undertaken on the Sun SAM-Remote client at the same time that recycling is occurring on the Sun SAM-Remote server. You should manually recycle on a time-interval basis that meets the needs of your site. Recycling in this manner takes more effort. However, this is the only way to ensure that data is well protected against relabeling cartridges incorrectly.

▼ To Recycle partially full VSNs

The VSNs for which a partially full status is reported can also be recycled.

1. **Examine the `recycler.log` file from the client to see if there are any partially full VSNs.**

Using the example in this chapter, you can consider VSN 000025 from the client, `zeke`, for recycling because its status is partially full. You can determine this from CODE EXAMPLE 3-13, which shows the client `recycler.log` file on `zeke`.

CODE EXAMPLE 3-13 The `recycler.log` File on Client `zeke`

```
# From the client zeke recycler.log file:
-----Status-----      ---Archives---      -----Percent-----      defaultset.2
Count      Bytes      Use Obsolete Free      Library:Type:VSN
no-data VSN      0      0      0      100      0      skyrs:at:000029
no-data VSN      0      0      0      99      1      skyrs:at:000026
partially full      111      2.8G      6      88      6      skyrs:at:000025
empty VSN      0      0      0      0      100      skyrs:at:000028
empty VSN      0      0      0      0      100      skyrs:at:000027
```


VSN 000025 shows that 6 percent of its space is in use. These are active archive images that must be rearchived before this VSN can be recycled. The following steps in this process show how to ensure that these active archive images are rearchived to another VSN.

2. Examine the `recycler.log` file from the server side to ensure that no active data from the server is archived on that VSN.

For example, look at the data for VSN 000025 in CODE EXAMPLE 3-14 that was selected for recycling from the previous step. The server's `recycler.log` file indicates that VSN 000025 is 6 percent free, which is the same percentage that was reported in the client's `recycler.log` file. The server is not aware of the client's archive images, so the server cannot report that the percent occupied is divided into 6 percent in-use archive images and 88 percent obsolete images. The server reports that all of the remaining 94 percent is consumed by obsolete archive images.

CODE EXAMPLE 3-14 The `recycler.log` File on Server sky

```
# From the Server log file:
```

-----Status-----	---Archives---		-----Percent-----			adic1 Library:Type:VSN
	Count	Bytes	Use	Obsolete	Free	
no-data VSN	0	0	0	100	0	adic1:at:000011
no_recycle VSN	0	0	0	100	0	adic1:at:000029
no-data VSN	0	0	0	100	0	adic1:at:000013
no-data VSN	0	0	0	100	0	adic1:at:000012
no_recycle VSN	0	0	0	99	1	adic1:at:000026
no_recycle VSN	0	0	0	94	6	adic1:at:000025
no_recycle VSN	0	0	0	51	49	adic1:at:000020
no-data VSN	0	0	0	17	83	adic1:at:000017
no-data VSN	0	0	0	10	90	adic1:at:000018
empty VSN	0	0	0	0	100	adic1:at:CLN003
no_recycle VSN	0	0	0	0	100	adic1:at:000021
no_recycle VSN	0	0	0	0	100	adic1:at:000022
no_recycle VSN	0	0	0	0	100	adic1:at:000027
no_recycle VSN	0	0	0	0	100	adic1:at:000028
no_recycle VSN	0	0	0	0	100	adic1:at:000023
no_recycle VSN	0	0	0	0	100	adic1:at:000024
empty VSN	0	0	0	0	100	adic1:at:000016
empty VSN	0	0	0	0	100	adic1:at:CLN001
empty VSN	0	0	0	0	100	adic1:at:CLN002
empty VSN	0	0	0	0	100	adic1:at:CLN004
partially full	12	88.3k	0	0	100	adic1:at:000000

3. Use the `chmed(1M)` command with the `+c` option on the VSN.

For the example in this procedure, the command is as follows:

```
server# chmed +c at.000025
```

This command indicates to the recycler that you want to rearchive the active files on this VSN. The files to be rearchived constitute the 6 percent as reported by the client's `recycler.log` file in the `Use` column. For more information about the `chmed(1M)` command, see the `chmed(1M)` man page.

4. Use the `sam-recycler(1M)` command to run the recycler again.

For the example in this procedure, the command is as follows:

```
client# sam-recycler -dvx
```

This marks each active file to be rearchived and indicates to the archiver that each active file should be rearchived to another VSN.

5. Start the archiver.

You can do this by either letting the archiver run normally, or by typing `:arrun` from the `samu(1M)` utility on the client to start the archiver. For more information about the `:arrun` command, see the `samu(1M)` man page or see the *Sun SAM-FS and Sun SAM-QFS Storage and Archive Management Guide*.

6. When archiving is complete, issue the `sam-recycler(1M)` command to rerun the recycler on the client.

This ensures that all active files have been rearchived.

For the example in this procedure, the command is as follows:

```
client# sam-recycler -dvx
```

7. (Optional) Use the `tplabel(1M)` or `odlabel(1M)` command to relabel the VSN from the server.

If the `Count`, `Bytes`, and `Use` fields are all 0 (zero), you can relabel the VSN from the server.

For the example in this procedure, you can use the following command to relabel the tape VSN:

```
server# tplabel -vsn 000025 -old 000025 at.000025
```

The preceding command relabels the VSN and destroys all data on the VSN. After this VSN is relabeled, you regain 88 percent of the space on this VSN.

If the media had been a magneto-optical disk, you would have used the `odlabel(1M)` command. For more information about the `odlabel(1M)` command, see the `odlabel(1M)` man page.

8. Devise a recycling schedule.

In a Sun SAM-FS or Sun SAM-QFS environment in which Sun SAM-Remote software is not enabled, you can create a `cron(1)` job so recycling is performed automatically. If Sun SAM-Remote software is enabled, however, do not automate the recycler.



Caution – It is very important that recycling activities not be undertaken on the Sun SAM-Remote client at the same time that recycling is occurring on the Sun SAM-Remote server. You should manually recycle on a time-interval basis that meets the needs of your site. Recycling in this manner takes more effort. However, this is the only way to ensure that data is well protected against relabeling cartridges incorrectly.

Recycling in a Sun SAM-Remote Environment—Method 2

This section presents another way you can recycle volumes using Sun SAM-remote software.



Caution – Use the the recycler in a Sun SAM-Remote environment only after following the steps in this procedure completely and only after testing your configuration to see that correct recycling is taking place.

▼ To Configure Recycling—Method 2

The steps for recycling volumes on the Sun SAM-Remote client are as follows.

1. **On the Sun SAM-Remote client, issue the `sam-recycler(1M)` command to determine which volumes are the best candidates for recycling.**

For example:

```
client# sam-recycler -dvx
```

You can determine this by analyzing the recycler log file.

2. **On the Sun SAM-Remote server, issue the `chmed(1M)` command to set the recycle flag on the chosen VSNs.**

For example:

```
server# chmed +c at.00025
```

3. **On the Sun SAM-Remote client, issue the `sam-recycler(1M)` command to recycle the chosen VSNs on the Sun SAM-Remote client.**

For example:

```
client# sam-recycler -dvx
```

4. **Wait until the VSNs being recycled are drained completely of archive images.**

The archiver on the client side does this.

5. **On the Sun SAM-Remote server, issue the `tplabel(1M)` or `odlabel(1M)` command to relabel the volumes after they are completely drained of archive images.**

6. **On the Sun SAM-Remote server, clear any flags that prevent the volumes from being used for archiving on the Sun SAM-Remote client (such as `R` or `C`).**

Again, it is very important that these recycling activities not be undertaken on the Sun SAM-Remote client at the same time that you are recycling volumes on the Sun SAM-Remote server.

Glossary

A

- addressable storage** The storage space encompassing online, nearline, offsite, and offline storage that is user-referenced through a Sun QFS, Sun SAM-FS, or Sun SAM-QFS file system.
- archive media** The media to which an archive file is written. Archive media can be removable tape or magneto-optical cartridges in a library. In addition, archive media can be a mount point on another system.
- archive storage** Copies of file data that have been created on archive media.
- archiver** The archive program that automatically controls the copying of files to removable cartridges.
- audit (full)** The process of loading cartridges to verify their VSNs. For magneto-optical cartridges, the capacity and space information is determined and entered into the automated library's catalog.
- automated library** A robotically controlled device designed to automatically load and unload removable media cartridges without operator intervention. An automated library contains one or more drives and a transport mechanism that moves cartridges to and from the storage slots and the drives.

B

backup storage A snapshot of a collection of files for the purpose of preventing inadvertent loss. A backup includes both the file's attributes and associated data.

block allocation map A bitmap representing each available block of storage on a disk and indicating whether the block is in use or free.

block size See DAU.

C

cartridge A physical entity that contains media for recording data. A tape or optical disk. Sometimes referred to as *a piece of media*, *a volume*, or *the medium*.

catalog A record of the VSNs in an automated library. There is one catalog for each automated library, and at a site, there is one historian for all automated libraries.

client-server The model of interaction in a distributed system in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called the client. The program satisfying the response is called the server.

connection The path between two protocol modules that provides reliable stream delivery service. A TCP connection extends from a TCP module on one machine to a TCP module on the other.

D

data device For a Sun QFS, Sun SAM-FS, or Sun SAM-QFS file system, a device or group of devices upon which file data is stored.

DAU (disk allocation unit)	The basic unit of online storage. Also called block size. The Sun SAM-FS and Sun SAM-QFS file systems support both a small and a large DAU. The small DAU is 4 kilobytes (2 ¹⁴ or 4096 bytes). The large DAU is 16, 32, or 64 kilobytes. The available DAU size pairs are 4/16, 4/32, and 4/64. In addition, the Sun QFS and Sun SAM-QFS file systems support a fully adjustable DAU, sized from 16 kilobytes through 65,528 kilobytes. The DAU you specify must be a multiple of 8 kilobytes.
device logging	A configurable feature that provides device-specific error information used to analyze device problems.
device scanner	Software within the Sun SAM-FS or Sun SAM-QFS file system that periodically monitors the presence of all manually mounted removable devices and that detects the presence of mounted cartridges that can be requested by a user or other process.
direct access	A file attribute (stage never) designating that a nearline file can be accessed directly from the archive media and need not be retrieved to disk cache.
direct-attached library	An automated library connected directly to a server using a SCSI interface. A SCSI attached library is controlled directly by the Sun SAM-FS or Sun SAM-QFS software by using the SCSI standard for automated libraries.
direct I/O	An attribute used for large block-aligned sequential I/O. The <code>setfa(1)</code> command's <code>-D</code> option is the direct I/O option. It sets the direct I/O attribute for a file or directory. If applied to a directory, the direct I/O attribute is inherited.
directory	A file data structure that points to other files and directories within the file system.
disk allocation unit	See DAU.
disk buffer	When using Sun SAM-Remote software, the disk buffer is a buffer on the server system that is used when archiving data from the client to the server.
disk cache	The disk-resident portion of the Sun SAM-FS and Sun SAM-QFS file system software. It is used to create and manage data files between online disk cache and archive media. Individual disk partitions or an entire disk can be used as disk cache.
disk space thresholds	An administrator-defined amount of disk space that is available to a user. This defines the range of desirable disk cache utilization. The high threshold indicates the maximum level of disk cache utilization. The low threshold indicates the minimum level of disk cache utilization. The releaser controls disk cache utilization based on these predefined disk space thresholds.

disk striping The process of recording a file across several disks, thereby improving access performance and increasing overall storage capacity. Also see entries for striping.

drive A mechanism for transferring data to and from a removable media volume.

E

Ethernet A local-area, packet-switched network technology. Originally designed for coaxial cable, it is now found running over shielded, twisted-pair cable. Ethernet is a 10- or 100-megabytes-per-second LAN.

extent array The array within a file's inode that defines where each data block assigned to the file is located on the disk.

F

family device set See family set.

family set A storage device that is represented by a group of independent physical devices, such as a collection of disks or the drives within an automated library. Also see disk cache family set.

FDDI Fiber distributed data interface. A 100-megabytes-per-second fiber-optic LAN.

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.

fibre-distributed data interface See FDDI.

file system A hierarchical collection of files and directories.

file system specific directives Archiver and releaser directives that follow global directives, are specific to a particular file system, and begin with `fs =`. File system specific directives apply until the next `fs =` directive line or until the end of file is encountered. If multiple directives affect a file system, the file system-specific directives override the global directives.

FTP File Transfer Protocol. An internet protocol for transferring files between two hosts over a TCP/IP network.

G

- global directives** Archiver and releaser directives that apply to all file systems and that appear before the first `fs =` line.
- grace period** For disk quotas, this is the amount of time that can elapse during which a user is allowed to create files and/or allocate storage after a user reaches their soft limit.

H

- hard limit** For disk quotas, a maximum limit on file system resources (blocks and inodes) that users cannot exceed.

I

- indirect block** A disk block that contains a list of storage blocks. The Sun QFS, Sun SAM-FS, and Sun SAM-QFS file systems have up to three levels of indirect blocks. A first-level indirect block contains a list of blocks used for data storage. A second-level indirect block contains a list of first-level indirect blocks. A third-level indirect block contains a list of second-level indirect blocks.
- inode** Index node. A data structure used by the file system to describe a file. An inode describes all the attributes associated with a file other than the name. The attributes include ownership, access, permission, size, and the file location on the disk system.
- inode file** A special file (`.inodes`) on the file system that contains the inode structures for all files resident in the file system. All Sun QFS, Sun SAM-FS, and Sun SAM-QFS inodes are 512 bytes long. The inode file is a metadata file, which is separated from file data in the Sun QFS and Sun SAM-QFS file systems.

K

kernel The central controlling program that provides basic system facilities. The UNIX kernel creates and manages processes, provides functions to access the file system, provides general security, and supplies communication facilities.

L

LAN Local area network.

lease In a Sun QFS shared file system, a lease grants a client host permission to perform an operation on a file for as long as the lease is valid. The metadata server issues leases to each client host. The leases are renewed as necessary to permit continued file operations.

library See automated library.

library catalog See catalog.

LUN Logical unit number.

M

mcf Master configuration file. The file that is read at initialization time that defines the relationships between the devices (the topology) within a Sun QFS, Sun SAM-FS, and Sun SAM-QFS environment.

media Tape or optical disk cartridges.

media recycling The process of recycling or reusing archive media with low use (that is, archive media with few active files).

metadata Data about data. Metadata is the index information needed to locate the exact data position of a file on a disk. It consists of information about files, directories, access control lists, symbolic links, removable media, segmented files, and the indexes of segmented files. Metadata must be protected because if data is lost, the metadata that locates the data must be restored before the lost data can be retrieved.

- metadata device** A separate device (for example, a solid-state disk or mirrored device) upon which Sun QFS and Sun SAM-QFS file system metadata is stored. Separating file data from metadata can increase performance. In the `mcf` file, a metadata device is declared as an `mm` device within an `ma` file system.
- mirror writing** The process of maintaining two copies of a file on disjointed sets of disks to prevent loss from a single disk failure.
- mount point** The directory on which a file system is mounted.
- multireader file system** The Sun QFS multireader file system is a single-writer, multireader capability that enables you to specify a file system that can be mounted on multiple hosts. Multiple hosts can read the file system, but only one host can write to the file system. Multiple readers are specified with the `-o reader` option on the `mount(1M)` command. The single-writer host is specified with the `-o writer` option on the `mount(1M)` command. For more information on the `mount(1M)` command, see the `mount_samfs(1M)` man page.

N

- name space** The metadata portion of a collection of files that identifies the file, its attributes, and its storage locations.
- nearline storage** Removable media storage that requires robotic mounting before it can be accessed. Nearline storage is usually less expensive than online storage, but it incurs a somewhat longer access time.
- network-attached automated library** A library, such as those from StorageTek, ADIC/Grau, IBM, or Sony, that is controlled using a software package supplied by the vendor. The Sun SAM-FS and Sun SAM-QFS file systems interface with the vendor software using a Sun SAM-FS or Sun SAM-QFS media changer daemon designed specifically for the automated library.
- NFS** Network file system. A Sun distributed file system that provides transparent access to remote file systems on heterogeneous networks.
- NIS** The SunOS 4.0 (minimum) Network Information Service. A distributed network database containing key information about the systems and the users on the network. The NIS database is stored on the master server and all the slave servers.

O

- offline storage** Storage that requires operator intervention for loading.
- offsite storage** Storage that is remote from the server and is used for disaster recovery.
- online storage** Storage that is immediately available (for example, disk cache storage).

P

- partition** A portion of a device or a side of a magneto-optical cartridge.
- preallocation** The process of reserving a contiguous amount of space on the disk cache for writing a file. This ensures that the space is contiguous. Preallocation can be performed only on zero-sized files. That is, the `setfa -l` command can be specified only for a file that is size zero. For more information, see the `setfa(1)` man page.
- prioritizing preview requests** Assigning priority to archive and stage requests that cannot be immediately satisfied.
- pseudo device** A software subsystem or driver with no associated hardware.

Q

- quota** The amount of system resources that a user is allowed to consume. Quotas are not supported for removable media or disk archive resources.

R

- RAID** Redundant array of inexpensive/independent disks. A disk technology that uses several independent disks to reliably store files. It can protect against data loss from a single disk failure, can provide a fault-tolerant disk environment, and can provide higher throughput than individual disks.

- recycler** A Sun SAM-FS and Sun SAM-QFS utility that reclaims space on cartridges that is occupied by expired archive copies.
- release priority** A method of calculating the release priority of a file within a file system by multiplying various weights by the corresponding file properties and then summing the results.
- releaser** A Sun SAM-FS and Sun SAM-QFS component that identifies archived files and releases their disk cache copies, thus making more disk cache space available. The releaser automatically regulates the amount of online disk storage to high and low thresholds.
- remote procedure calls** See RPC.
- removable media file** A special type of user file that can be accessed directly from where it resides on a removable media cartridge, such as magnetic tape or optical disk cartridge. also used for writing archive and stage file data.
- robot** The portion of an automated library that moves cartridges between storage slots and drives. Also called a transport.
- round robin** A data access method in which entire files are written to logical disks in a sequential fashion. When a single file is written to disk, the entire file is written to the first logical disk. The second file is written to the next logical disk, and so on. The size of each file determines the size of the I/O.
- By default, Sun QFS, Sun SAM-FS, and Sun SAM-QFS file systems implement striped data access unless striped groups are present. Files are round-robined if round robin access is specified. If the file system contains mismatched striped groups, striping is not supported and round robin is forced.
- Also see glossary entries for disk striping and striping.
- RPC** Remote procedure calls. The underlying data exchange mechanism used by NFS to implement custom network data servers.

S

- samfsdump** A program that creates a control structure dump and copies all the control structure information for a given group of files. It is analogous to the UNIX `tar(1)` utility, but it does not generally copy file data.
- samfsrestore** A program that restores inode and directory information from a control structure dump.

- SCSI** Small Computer System Interface. An electrical communication specification commonly used for peripheral devices such as disk and tape drives and automated libraries.
- small computer system interface** See SCSI.
- soft limit** For disk quotas, a threshold limit on file system resources (blocks and inodes) that you can temporarily exceed. Exceeding the soft limit starts a timer. When you exceed the soft limit for the specified time (default is one week), no further system resources can be allocated until you reduce file system use to a level below the soft limit.
- staging** The process of copying a nearline or offline file from archive storage back to online storage.
- storage family set** A set of disks that are collectively represented by a single disk family device.
- storage slots** Locations inside an automated library in which cartridges are stored when not being used in a drive. If the library is direct-attached, the contents of the storage slots are kept in the automated library's catalog.
- stripe size** The number of disk allocation units (DAUs) to allocate before moving to the next device of a stripe. If `stripe=0`, the file system uses round-robin access, not striped access.
- striped group** A collection of devices within a Sun QFS or Sun SAM-QFS file system and defined in the `mcf` file as one (usually two) or more `gXXX` devices. Striped groups are treated as one logical device and are always striped with a size equal to the disk allocation unit (DAU). You can specify up to 128 striped groups within a file system, but you can specify no more than 252 total devices.
- striping** A data access method in which files are simultaneously written to logical disks in an interlaced fashion. All Sun QFS, Sun SAM-FS, and Sun SAM-QFS file systems enable you to declare either striped or round robin access for each individual file system. The Sun QFS and Sun SAM-QFS file systems enable you to declare striped groups within each file system. Also see the glossary entry for round robin.
- Sun SAM-FS** The Sun Storage and Archive Manager File System. The Sun SAM-FS software controls the access to all files stored and all devices configured in the master configuration file (`mcf`).
- Sun SAM-QFS** The Sun SAM-QFS software combines the Sun Storage and Archive Manager with the Sun QFS file system. Sun SAM-QFS offers a high-speed, standard UNIX file system interface to users and administrators in conjunction with the storage and archive management utilities. It uses many of the commands available in the Sun SAM-FS command set as well as standard UNIX file system commands.

Sun SAM-Remote client	A Sun SAM-Remote client is a Sun SAM-FS or Sun SAM-QFS system that establishes a Sun SAM-Remote client daemon that contains a number of pseudodevices. It might or might not have its own library devices. The client depends on a Sun SAM-Remote server for archive media for one or more archive copies.
Sun SAM-Remote server	The Sun SAM-Remote server is both a full-capacity Sun SAM-FS or Sun SAM-QFS storage management server and a Sun SAM-Remote server daemon that defines libraries to be shared among Sun SAM-Remote clients.
superblock	A data structure in the file system that defines the basic parameters of the file system. It is written to all partitions in the storage family set and identifies the partition's membership in the set.



T

tar	Tape archive. A standard file/data recording format used by the Sun SAM-FS and Sun SAM-QFS software for archive images.
TCP/IP	Transmission Control Protocol/Internet Protocol. The internet protocols responsible for host-to-host addressing and routing, packet delivery (IP), and reliable delivery of data between application points (TCP).
thresholds	A mechanism for defining the desirable available storage window for online storage. Thresholds set the storage goals for the releaser. Also see disk space thresholds.
timer	Quota software that keeps track of the time elapsed between a user reaching a soft limit and a hard limit being imposed on the user.



V

volume	A named area on a cartridge for sharing data. A cartridge has one or more volumes. Double-sided cartridges have two volumes, one on each side.
volume overflow	A capability that enables the system to span a single file over multiple volumes. Volume overflow is useful for sites using very large files that exceed the capacity of their individual cartridges.

VSN Volume serial name. If you are archiving to removable media cartridges, the VSN is a logical identifier for magnetic tape and optical disk that is written in the volume label. If you are archiving to disk cache, this is the unique name for the disk archive set.

W

WORM Write once read many. A storage classification for media that can be written only once but read many times.

Index

SYMBOLS

/etc/opt/SUNW/samfs/archiver.cmd

see archiver.cmd

/etc/opt/SUNWsamfs/mcf file

see mcf file

/etc/opt/SUNWsamfs/recycler.cmd file

see recycler.cmd file

A

archiver(1M) command, 22

archiver.cmd file, 28, 30

Archiving with Sun SAM-Remote, 6

C

Catalog, 21

chmed(1M) command, 23, 27, 48, 50

Client configuration file, 13

Client configuration overview, 5

Commands

archiver(1M), 22

chmed(1M), 23, 27, 48, 50

cron(1), 46, 49

odlabel(1M), 48, 50

pkginfo(1M), 9

samcmd(1M), 11

samd(1M), 19

sam-recycler(1M), 32, 36, 48, 50

samu(1M), 48

tplabel(1M), 48, 50

Configuration, 7

cron(1) command, 46, 49

D

Directives for recycling, 29

Drives, idling, 11

I

Installation, 7

L

Log file for the recycler, 32, 36, 44

M

mcf file, 4, 12, 13, 14, 25, 26

N

NFS, 2

no-data VSNs, 44

O

odlabel(1M) command, 48, 50
Overview, 3

P

partially full VSNs, 46
pkginfo(1M) command, 9

R

-recycle_hwm directive, 29
-recycle_ignore directive, 29
-recycle_mingain directive, 29
-recycle_vsncount directive, 29
Recycler log file, 32, 36, 44
recycler.cmd file, 23, 30, 31
recycler.sh script, 32
Recycling with Sun SAM-Remote, 23

S

samcmd(1M) command, 11
samd(1M) command, 19
sam-recycler(1M) command, 32, 36, 48, 50
samu(1M) command, 48
Server configuration file, 15
Server configuration overview, 4

T

tplabel(1M) command, 48, 50