

IBM TotalStorage SAN File System
(based on IBM Storage Tank™ technology)



Administrator's Guide and Reference

Version 2 Release 1

IBM TotalStorage SAN File System
(based on IBM Storage Tank[™] technology)



Administrator's Guide and Reference

Version 2 Release 1

Note

Before using this information and the product it supports, read the information in "Notices."

Second Edition (June 2004)

This edition applies to the IBM TotalStorage SAN File System and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office servicing your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for reader's comments is provided at the back of this publication. If the form has been removed, you may address your comments to:

International Business Machines Corporation
Design & Information Development
Department CGFA
PO Box 12195
Research Triangle Park, NC 27709-9990
U.S.A.

You can also submit comments by selecting Feedback at www.ibm.com/storage/support/.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2003, 2004. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this guide. ix

Who should use this guide ix

Notices in this guide ix

Publications x

 SAN File System publications. x

 SAN File System related publications xi

Web sites xi

Summary of changes xi

Chapter 1. Getting started 1

Overview of SAN File System 1

 What is IBM TotalStorage SAN File System?. 1

 What are the major features? 2

 What's in the box? 3

 Where do I start? 3

 How can I get quick help with problems? 4

Concepts 4

 Administrative security. 4

 Administrative server 6

 Alerts and events. 6

 Backup and restore 8

 Clients 13

 Cluster 22

 Common Information Model 25

 Components 25

 Container 27

 Data migration 27

 Engines. 29

 Filesets 30

 FlashCopy images 34

 Global namespace 37

 Locks and leases. 39

 Logs and traces 40

 Master console 43

 Metadata server 44

 Remote access 46

 Service alert 46

 SNMP 47

 Storage management 48

 Storage subsystems. 50

 Storage pools. 51

 User interfaces 53

 Volumes 54

Chapter 2. Accessing the administrative interfaces 57

Accessing the administrative command-line interface 57

Accessing the SAN File System console 57

Chapter 3. Monitoring 59

Creating a report 59

Monitoring metadata server HBA statistics 59

Viewing metadata server system statistics 59

Viewing SAN File System component statistics 60

Viewing system status. 60

Chapter 4. Managing 61

Managing administration properties 61

 Changing LDAP configuration settings 61

 Changing RSA userid and password 61

 Viewing administration properties details 61

Managing alerts and logs. 62

 Adding SNMP managers 62

 Clearing logs 62

 Deleting SNMP managers 62

 Modifying SNMP managers 63

 Modifying SNMP traps 63

 Setting up SNMP traps 63

 Viewing logs 63

 Viewing SNMP managers. 64

Managing backups 64

 Backing up using the LUN method 64

 Backing up using the file-based (API) method 65

Managing clients 67

 Accessing a client through SSH. 67

 Accessing a client through telnet 67

 Accessing a client through a remote console utility 68

 Displaying the client-driver version 68

 Downloading client software 69

 Enabling client awareness of LUN expansion 69

 Granting root privileges to a client 69

 Implementing file sharing - UNIX fileset on a Windows client 69

 Implementing file sharing - Windows fileset on a UNIX client 70

 Listing client sessions 70

 Listing clients that can see a LUN 70

 Listing clients with root privileges. 70

 Revoking root privileges from a client 70

 Setting administrative privileges and fileset permissions on UNIX 71

 Setting administrative privileges and fileset permissions on Windows. 71

 Starting a client 72

 Stopping a client 74

 Viewing client-session details 75

 Viewing client-session statistics. 75

Managing the cluster 75

 Changing active cluster states 76

 Configuring the cluster language 76

 Configuring cluster tuning 76

 Listing cluster configuration. 76

 Starting the cluster 77

 Stopping the cluster 77

 Upgrading the cluster 77

 Viewing cluster details. 77

 Viewing cluster statistics 78

 Viewing the cluster software version 78

Managing disaster recovery	78	Starting a metadata server	99
Creating a recovery file	78	Starting the metadata server restart service	99
Creating recovery scripts	78	Stopping a metadata server.	100
Deleting a recovery file	79	Stopping the metadata check	100
Listing recovery files	79	Stopping the metadata server restart service	100
Restoring the master console	79	Tuning the metadata server restart service.	101
Restoring the engine hardware and operating system	79	Viewing metadata server details	101
Restoring SAN connectivity	80	Viewing metadata server statistics	101
Restoring SAN File System software	81	Viewing metadata server restart service statistics	101
Restoring SAN File System cluster configuration	81	Viewing metadata server restart service tuning details.	102
Restoring SAN File System metadata	83	Removing down-level metadata server software	102
Restoring SAN File System clients	84	Managing policies	102
Restoring SAN File System user data	84	Activating a policy	103
Managing engines	84	Changing the rules in a policy.	103
Accessing an engine through SSH	85	Checking the placement of current files.	103
Accessing the RSA II adapter	85	Copying a policy	103
Collecting diagnostic data on engines.	86	Creating a policy	104
Listing engines	87	Deleting a policy	104
Powering on the engine	87	Listing policies	104
Powering on the engine	87	Changing the name of a policy	104
Restarting the engine	87	Viewing policy details	105
Starting the Configuration/Setup Utility.	88	Viewing policy rules	105
Viewing engine fan status	88	Viewing policy statistics.	105
Viewing engine power status	88	Managing processes	105
Viewing engine statistics	89	Listing processes	105
Viewing the engine status summary	89	Viewing process details	105
Viewing engine temperatures	89	Viewing process limits	105
Viewing engine timeouts	89	Managing storage pools	106
Viewing engine voltage status	89	Changing storage pool settings	106
Viewing vital engine data.	89	Creating a storage pool	106
Managing filesets	90	Deleting a storage pool	107
Attaching a fileset	90	Disabling the default storage pool	107
Changing fileset settings	90	Listing storage pools	107
Creating a fileset	91	Listing volumes in a storage pool	107
Deleting a fileset.	91	Setting the default storage pool	107
Detaching a fileset	91	Viewing storage pool details	108
Listing filesets	92	Viewing storage pool statistics.	108
Reassigning a fileset to a different metadata server	92	Managing users	108
Viewing fileset details	92	Granting SAN File System user access and roles	108
Viewing fileset placement information	92	Listing user roles	108
Viewing fileset settings	92	Listing users.	108
Viewing fileset statistics	93	Timing out all user authorizations	109
Managing FlashCopy images	93	Managing volumes and LUNs.	109
Creating a FlashCopy image.	93	Activating a volume	109
Deleting a FlashCopy image.	93	Adding LUNs to SAN File System	109
Listing FlashCopy images	93	Adding volumes to a storage pool	110
Reverting to a previous FlashCopy image	93	Changing volume settings	110
Viewing FlashCopy image details	94	Increasing volume size	110
Managing metadata servers	94	Listing available LUNs	111
Remotely accessing the master console	94	Listing files on a volume	111
Accessing the administrative server through a browser	96	Listing LUNs	111
Adding a metadata server to the cluster	96	Listing volumes	111
Changing the master metadata server	97	Managing free and allocated storage space	111
Changing metadata server hostname	98	Removing volumes from a storage pool	112
Changing metadata server IP address.	98	Suspending a volume.	112
Checking metadata	98	Viewing available LUN details	113
Listing metadata servers	99	Viewing LUN details	113
Removing a metadata server from the cluster	99	Viewing LUN statistics	113
		Viewing volume details	113
		Viewing volume settings.	113

Viewing volume statistics	113
Appendix A. Accessibility	115
Appendix B. Commands.	117
Administrative CLI overview	117
Administrative command-line interface	118
Command modes	118
Naming guidelines	119
Standard format parameters	120
Standard help parameters	122
Standard listing parameters	122
Syntax diagram conventions	123
Common commands	125
tankpasswd	129
Administrative commands	130
activatevol	135
addprivclient	136
addserver	137
addsnmpmgr	137
attachfileset	138
autofilesetserver	140
buildscript	141
catlog	142
catpolicy	144
chadvclusterconfig	144
chclusterconfig	146
chfileset	148
chldapconfig	150
chpool	152
chvol	153
clearlog	154
collectdiag	155
detachfileset	156
disabledefaultpool	157
dropserver	158
expandvol	159
legacy	160
lsadmuser	167
lsautorestart	169
lsclient	173
lsdrfile	176
lsengine	178
lsfileset	181
lsimage	184
lslun	187
lspolicy	190
lspool	192
lsproc	195
lsserver	197
lssnmpmgr	201
lstrapsetting	201
lsvol	202
mkfileset	205
mkdrfile	207
mkimage	208
mkpolicy	209
mkpool	211
mkvol	212
quiescecluster	214

rediscoverluns	215
reportclient	216
reportfilesetuse	216
reportvolfiles	217
resetadmuser	218
restartengine	219
resumecluster	220
reverttoimage	220
rmfileset	222
rmdrfile	223
rmimage	223
rmpolicy	224
rmpool	225
rmprivclient	226
rmsnmpmgr	227
rmvol	227
setautorestart	229
setdefaultpool	230
setfilesetserver	231
setmaster	232
setport	235
settrap	236
startautorestart	238
startcluster	239
startengine	240
startmetadatacheck	240
startserver	242
statcluster	243
statengine	246
statfile	249
statfileset	252
statldap	253
statpolicy	254
statserver	256
stopautorestart	258
stopcluster	258
stopengine	259
stopmetadatacheck	260
stopserver	261
suspendvol	262
upgradecluster	262
usepolicy	263
Client commands	264
Common client commands	264
AIX-client commands	268
Linux-client commands	278
Solaris-client commands	287
Windows-client command	295
SAN File System installation commands	295
setupsfs	295
setupstclient	297
tmvt	298
Service commands and utilities	299
disableConsoleTrace	301
enableConsoleTrace	301
mktruststore	302
obdc	303
startCimom	307
startConsole	308
stfsdebug	308
stfsstat	309

stopCimom	312
stopConsole	313
tank extractbootrecord	313
tank lscluster	314
tank lsversion	316
tank lsdisklabel	317
tank resetcluster	319
tank resetversion	319

Appendix C. Environment variables 321

Appendix D. File placement policy syntax 323

Appendix E. Getting help, service, and information 329

Before you call for service	329
Getting help online	329
Getting help by telephone	330

Appendix F. Logs. 331

Administrative log	331
Audit log.	332
Event log.	333
Security log	333
Server log	334

Appendix G. SAN File System console panels 335

Add Server to the Cluster panel	335
Add Volumes to a Storage Pool - Add Volumes panel	335
Add Volumes to a Storage Pool - Introduction panel	335
Add Volumes to a Storage Pool - Select Client panel	336
Add Volumes to a Storage Pool - Select Storage Pool panel	336
Add Volumes to a Storage Pool - Verify Settings panel	337
Add Volumes to a Storage Pool - Volume Settings panel	337
Administration Properties – Details panel	337
Administration Properties – LDAP Access panel	338
Administrative Properties – System Access panel	339
Administrative Log panel	339
Attach Fileset panel	340
Audit Log panel	341
Available Data LUNs panel.	342
Available Metadata LUNs panel	344
Change Power State of Engine panel	345
Change State of Cluster panel	345
Check Metadata panel	346
Check Metadata Progress panel	347
Client Sessions panel	348
Clients that can see the LUN panel	349
Cluster panel	349
Cluster Log panel	351
Cluster Properties – Details panel	352

Cluster Properties – Localization panel	354
Cluster Properties – Tuning panel	354
Collect Diagnostic Data panel	354
Collect Diagnostic Data Progress panel	355
Create a Fileset panel.	356
Create FlashCopy Images of Filesets - Introduction panel	357
Create FlashCopy Images of Filesets - Select Filesets panel	357
Create FlashCopy Images of Filesets - Set Properties panel	358
Create FlashCopy Images of Filesets - Verify Settings panel	359
Create a Policy - Add Rules panel	359
Create a Policy - Edit Rules panel	360
Create a Policy - High-Level Settings panel	360
Create a Policy - Introduction panel	361
Create Recovery File panel	361
Create a Storage Pool - Add Volumes panel	361
Create a Storage Pool - Introduction panel.	362
Create a Storage Pool - Pool Settings panel	362
Create a Storage Pool - Select Client panel.	363
Create a Storage Pool - Verify Settings panel	363
Create a Storage Pool - Volume Settings panel	364
Data LUNs panel	364
Delete Filesets panel	366
Delete FlashCopy Images panel	366
Delete Storage Pools panel	367
Detach Filesets panel	367
Details of Client Session panel.	367
Details of Image panel	369
Details of LUN panel.	369
Disaster Recovery panel.	371
Download Client Software panel	371
Engines panel	372
Engines Properties – Fans panel	373
Engines Properties – Power panel	374
Engines Properties – Summary panel	374
Engines Properties – Temperatures panel	376
Engines Properties – Timeouts panel	377
Engines Properties – Vital Engine Data panel.	378
Engines Properties – Voltages panel	378
Fileset Properties – Details panel	379
Fileset Properties – General Settings panel.	380
Fileset Properties – Quota Options panel	381
Filesets panel	382
FlashCopy Images panel.	383
General Properties of Storage Pools panel	383
Metadata LUNs panel	384
Placement of Fileset panel	385
Policies panel	386
Policy Properties – Details panel	386
Policy Properties – Rules panel	387
Policy Statistics – Per Pool panel	387
Policy Statistics – Per Rule panel	388
Privileged Clients panel	388
Processes panel.	389
Processes General Properties – Cluster-Level Details panel	389
Processes General Properties – Server-Level Details panel	390

Remove Volumes panel	390
Restart Service panel	391
Restart Service Details panel	392
Restart Service - Tuning panel	393
Restart Service Statistics – Probe Overview panel	393
Restart Service Statistics – Test Details panel	394
Revert Fileset to Image panel	394
Roles panel	395
Signon panel	396
Security Log panel	396
Server Properties – Details panel	397
Servers panel	398
SNMP Properties – SNMP Events panel	400
SNMP Properties – SNMP Managers panel	400
Statistics – Client Sessions panel	401
Statistics – Cluster panel	401
Statistics – Create Report panel	402
Statistics – Engines panel	403
Statistics – Filesets panel	404
Statistics – Report panel	404
Statistics – Storage Pools panel	405
Statistics – Servers panel	405
Statistics – Volumes panel	407
Stop Cluster panel	408
Stop Servers panel	408
Storage Pools panel	409
Storage Pool Properties – Available Data LUNs panel	410

Storage Pool Properties – Available Metadata LUNs panel	411
Storage Pool Properties – Current Volumes panel	411
Storage Pool Properties – Details panel	412
Storage Pool Properties – General Settings panel	413
System Overview panel	413
Users panel	417
Using the Help Assistant	417
Volume Properties – Details panel	419
Volume Properties – General Settings panel	419
Volume Properties – Size panel	419
Volume Remove Progress panel	420
Volumes panel	420

Appendix H. Prerequisites 423

Supported browsers	424
------------------------------	-----

Appendix I. Purchasing additional services. 425

Appendix J. Notices 427

Trademarks	428
----------------------	-----

Appendix K. Glossary 431

Index 437

About this guide

This topic informs the SAN File System administrator about the information contained within the *Administrator's Guide and Reference*.

This guide provides an overview of the IBM TotalStorage SAN File System and detailed administration instructions.

Who should use this guide

This topic identifies the intended audience for the *Administrator's Guide and Reference*.

This guide should be used by personnel assigned to monitor and manage SAN File System on a regular basis.

Depending on the specific role assigned within the system (that is, Monitor, Operator, Backup, and Administrator), SAN File System administrators should have experience in at least the following skills, or have access to personnel with experience in these skills:

- Microsoft® Windows® and Windows Advanced Server (depending on your client environment)
- AIX® (depending on your client environment)
- Linux system administration
- Lightweight Directory Access Protocol (LDAP) management
- Networking and network management
- SAN management
- Command-line interface scripting
- Critical business issues (such as backup, disaster recovery, and security)

Notices in this guide

This topic describes the notices in the *Administrator's Guide and Reference*.

These notices are used in this guide to convey specific meanings:

Note: These notices provide important tips, guidance, or advice.

Attention: These notices indicate possible damage to programs, devices, or data. An attention notice appears before the instruction or situation in which damage could occur.

CAUTION:

These notices indicate situations that can be potentially hazardous to you. A caution notice appears before the description of a potentially hazardous procedure step or situation.

DANGER

These notices indicate situations that can be potentially lethal or extremely hazardous to you. A danger notice appears before a description of a potentially lethal or extremely hazardous procedure step or situation.

Publications

This topic describes the publications in the SAN File System library and in related libraries.

SAN File System publications

This topic describes the publications in the SAN File System library.

The following publications are available in the SAN File System library. They are provided in softcopy on the *IBM TotalStorage SAN File System Publications CD* and at www.ibm.com/storage/support. To use the CD, insert it in the CD-ROM drive. If the CD does not launch automatically, follow the instructions on the CD label.

Note: The softcopy version of these publications are accessibility-enabled for the IBM® Home Page Reader.

- *IBM TotalStorage SAN File System Release Notes*

This document provides any changes that were not available at the time the publications were produced. This document is available only from the technical support Web site: www.ibm.com/storage/support

- *IBM TotalStorage SAN File System Software License Information*

This publication provides multilingual information regarding the software license for IBM TotalStorage SAN File System Software.

- *IBM TotalStorage SAN File System Administrator's Guide and Reference, GA27-4317*

This publication introduces the concept of SAN File System, and provides instructions for configuring, managing, and monitoring the system using the SAN File System console and administrative command-line interfaces. This book also contains a commands reference for tasks that can be performed at the administrative command-line interface or the command window on the client machines..

- *IBM TotalStorage SAN File System Basic Configuration for a Quick Start, GX27-4058*

The document walks you through basic SAN File System configuration and specific tasks that exercise basic SAN File System functions. It assumes that the physical configuration and software setup have already been completed.

- *IBM TotalStorage SAN File System Maintenance and Problem Determination Guide, GA27-4318*

This publication provides instructions for adding and replacing hardware components, monitoring and troubleshooting the system, and resolving hardware and software problems.

Note: This document is intended only for trained support personnel.

- *IBM TotalStorage SAN File System Installation and Configuration Guide, GA27-4316*

This publication provides detailed procedures to set up and cable the hardware, install and upgrade the SAN File System software, perform the minimum required configuration, and migrate existing data.

- *IBM TotalStorage SAN File System Messages Reference, GC30-4076*

This publication contains message description and resolution information for errors that can occur in the SAN File System software.

- *IBM TotalStorage SAN File System Planning Guide, GA27-4344*

This publication provides detailed procedures to plan the installation and configuration of SAN File System.

- *IBM TotalStorage SAN File System System Management API Guide and Reference, GA27-4315*

This publication contains guide and reference information for using the CIM Proxy API, including common and SAN File System-specific information.

Note: This document contains information and procedures intended for only selected IBM Business Partners. Contact your IBM representative before using this publication.

SAN File System related publications

These publications are related to SAN File System.

- *IBM TotalStorage® Subsystem Device Driver User's Guide, SC26-7637*

Web sites

This topic discusses any Web sites that offer additional, up-to-date information about SAN File System.

The following Web sites have additional information about SAN File System:

- www.ibm.com/storage/support
- www.ibm.com/storage/software/virtualization/sfs

Summary of changes

This section describes the enhancements made to SAN File System in version 2.1.0.

The following list describes the technical changes and enhancements made to SAN File System for version 2.1.0.

- **New Web address for the user interface** — The Web address that is used to access the SAN File System console has changed to:

`https://IP_address:7979/sfs`

- **Software-only solution** — SAN File System is now a software-only solution that leverages the strengths of the standard SuSE Linux Enterprise Server platform.
- **Heterogeneous SAN environment** — SAN File System now supports a wide variety of SAN configurations, which eases scaling to large numbers of storage devices and clients
- **Storage pool access** — The metadata servers require access to the metadata storage (system storage pool). They should not have access to the user data storage (user storage pools). Conversely, the clients require access to the user storage pool. They should not have access to the system storage pools.
- **Non-ASCII Unicode characters** — SAN File System supports both uppercase and lowercase non-ASCII Unicode characters in file names. SAN File System policies also support non-ASCII Unicode characters.
- **Additional client platforms** — SAN File System supports these additional client platforms:
 - AIX 5.2 (32-bit and 64-bit)

- Red Hat Enterprise Linux 3.0 Advanced Server
- Sun Solaris 9 (64-bit)

SAN File System also supports AIX and Solaris clients running in a clustered environment.

- **FIFO file-system objects** — SAN File System supports FIFO file-system objects in the global namespace for UNIX-based clients.
- **Opportunistic locks (oplocks)** — Windows-based clients can create and use opportunistic locks (oplocks). SAN File System supports level 1, 2, batch, and filter locks.
- **Application binary support** — Application binaries for UNIX-based clients use the POSIX definition of three sets of three file modes bits: one set of reach user, group, and other.
- **Additional storage subsystem** — SAN File System supports heterogeneous, simultaneously connected storage subsystems on clients with HBA sharing (subject to limitations of the client platform, drivers, and storage vendors). SAN File System conforms to FCP standards and is designed to work with any FCP-compliant storage subsystems for user data storage, including:
 - EMC Symmetrix
 - Hitachi 9900 Series and 9900V Series
 - IBM FAStT 600T, 700, and 900 running firmware version 8.4 and software version 8.41

Note: Only IBM storage subsystem are supported for the system storage pool. SAN File System supports an unlimited number of LUNs for user data storage. However, the amount of user data storage that you can have in your environment is determined by the amount of storage that is supported by the storage subsystems and the client operating systems.

- **High availability, non-disruptive maintenance, and serviceability** — SAN File System has been improved to provide greater availability of the cluster.
 - **Automatic workload failover (filesets and master role)** — SAN File System supports the non-disruptive, automatic failover of the workload. SAN File System automatically redistributes the filesets of a failed or manually stopped metadata server and, if necessary, reassigns the master role to another metadata server in the cluster. SAN File System uses automatic workload failover to provide non-disruptive maintenance for the metadata servers
 - **Automated fallback of workload** — If you statically assigned any filesets to a specific metadata server, SAN File System will automatically assign those filesets back to their statically assigned metadata server after the engine hosting the failed metadata server comes back online.
 - **Non-disruptive LUN additions** — SAN File System supports adding new LUNs to an existing system without restarting the system.
 - **Non-disruptive fileset movement** — You can manually reassign a fileset to another metadata server without disrupting service to the clients.
- **Master console improvements** — In addition to updated software levels, the master console allows you to configure Windows and IBM Director and to modify the machine name of the storage engines.
- **Serviceability improvements** — Improvements have been made to various serviceability components, including:
 - Metadata checker support of a lost-and-found directory for unreferenced objects.
 - One button data collection utility that has a consistent interface across all platforms.
 - Performance improvements that reduce the impact of logging and tracing.

- LUN and volume reporting to allow for easy comparison between the SAN configurations of each client.
- Target Machine Validation Tool (TMVT) hardware and software verification tool. The TMVT runs before the SAN File System setup script and provides a report of installed levels of hardware and software, as well as the required levels of hardware and software.
- IBM is providing you an optional script to customize your SuSE Linux Enterprise operating system. This script enables you to remove unnecessary packages from your system and to keep on your system only those packages required to run SAN File System. The script, named `ktl.sh`, can be found on the top level directory of the SAN File System Software CD.
- **Usability improvements** — SAN File System has many usability improvements, including:
 - A Refresh button is on many SAN File System console panels to enable you to update the panel with latest information.
 - The dashboard has been simplified to provide you with a quick summary of the entire system.
 - The message IDs in the SAN File System console logs are linked to the Information Center to help you quickly resolve problems.
 - The helps for messages have been improved.
 - You can filter the SAN File System console logs by message date and severity.
 - You can display statistics for fileset transactions, policy sets, and file metadata.
 - You can add and remove individual privileged clients from the administrative command-line interface and SAN File System console.
 - SAN File System supports additional configuration options for the metadata servers and cluster.
 - You can start the cluster and metadata servers in the offline state.
- **Terminology changes** — There are two major terminology changes in SAN File System:
 - The term *container* has been changed to the more intuitive term, *fileset*.
 - The administrative command-line interface tool changed from *tanktool* to *sfscli*.
 - The string *tank* in most references has been replaced with *sfs* or *sanfs*.

Chapter 1. Getting started

This section describes the basic information needed to understand and administer SAN File System.

Overview of SAN File System

This section provides a brief overview of SAN File System and describes what comes with SAN File System, where to start, and how to get help.

The remainder of the information in this guide is organized as follows:

- Chapter 3, “Monitoring,” on page 59 provides information about how to keep track of system performance through tasks such as creating a report, or viewing statistics.
- Chapter 4, “Managing,” on page 61 gives step-by-step instructions for performing ongoing administrative tasks for SAN File System through the management of alerts, clients, engines, users, volumes, and so forth.
- The appendices provide the following additional information:
 - Accessibility features of the SAN File System console and help system
 - Commands for the administrative command-line interface
 - Administrative command-line interface environment variables
 - Sources for getting additional help, service and information
 - Logs to assist in monitoring and troubleshooting
 - Information on using the SAN File System console and field descriptions for SAN File System console panels
 - Task prerequisites
 - Information about purchasing additional services
 - Notices

What is IBM TotalStorage SAN File System?

This topic provides a brief overview of IBM TotalStorage SAN File System.

IBM TotalStorage SAN File System is a storage area network (SAN)-based, scalable, and highly-available file system and storage management solution for file aggregation and concurrent data sharing in an open, multi-platform environment. It uses SAN technology, which allows an enterprise to connect a large number of heterogeneous computers and share a large number of heterogeneous storage devices over a high-performance network.

With SAN File System, heterogeneous clients can access shared data directly from large, high-performance, high-function storage systems, such as IBM TotalStorage Enterprise Storage Server[®] (ESS) and IBM TotalStorage SAN Volume Controller. The SAN File System is built on a Fibre Channel network, and is designed to provide superior I/O performance for data sharing among heterogeneous computers. It also provides growth capability and simplified storage management.

SAN File System differs from conventional distributed file systems in that it uses a data-access model that separates file metadata (information about the files, such as owner, permissions, and the physical file location) from actual file data (contents of

the files). The metadata is provided to clients by the metadata servers. Clients communicate with the metadata servers only to get the information they need to locate and access the files. Once they have this information, the SAN File System clients access data directly from storage devices through the clients' own direct connection to the SAN. Direct data access eliminates server bottlenecks and provides the performance necessary for data-intensive applications.

SAN File System presents a single, global namespace to clients where they can create and share data using uniform file names from any client or application. Data consistency and integrity are maintained through SAN File System's management of distributed *locks* and the use of *leases*. SAN File System provides locks that enable file sharing among SAN File System clients, and when necessary, provides locks that allow clients to have exclusive access to files. A lease determines the maximum period of time that a metadata server guarantees the locks that it grants to clients. A client must contact the metadata server before the lease period ends in order to retain its locks.

SAN File System also provides automatic file placement through the use of policies and rules. Based on the rules specified in a centrally-defined and managed policy, SAN File System automatically stores data on devices in *storage pools* that are specifically created to provide the capabilities and performance appropriate for how the data is accessed and used.

What are the major features?

This topic summarizes the major features of SAN File System.

Direct data access by exploitation of SAN technology

SAN File System uses a data-access model that allows client systems to access data directly from storage systems using a high-bandwidth SAN, without interposing servers. Direct data access helps eliminate server bottlenecks and provides the performance necessary for data-intensive applications.

Global namespace

SAN File System presents a single, uniform, global namespace view of all files in the system to all of the clients, without manual, client-by-client configuration by the administrator. A file can be identified using the same path and file name, regardless of the client system from which it is being accessed. The global namespace shared directly by clients also reduces the requirement of data replication. As a result, the productivity of the administrator as well as the users accessing the data is improved.

File sharing

All clients, regardless of operating system or hardware platform, have uniform access to the data stored (under the global namespace) in the system. File metadata, such as last modification time, are presented to users and applications in a form that is compatible with the native file system interface of the platform.

Policy based storage and data management

SAN File System is aimed at simplifying the storage-resource management and reducing the total cost of ownership by the policy-based automatic placement of files on appropriate storage devices. The storage administrator can define storage

pools depending on specific application requirements and quality of services, and define rules based on data attributes to store the files at the appropriate storage devices automatically. SAN File System provides the storage administrator with policy-based data management that automates the management of storage resources and the data stored on those resources.

What's in the box?

This topic describes what comes with SAN File System.

SAN File System is shipped with three CD-ROMs:

- *SAN File System Software CD* contains the SAN File System software that runs on the SUSE Linux Enterprise platform on prerequisite IBM xSeries[®] hardware (or equivalent), called *engines*. It also contains SAN File System client software that provides clients with local access to the global namespace on your SAN. The client software must be installed on client machines.
- *SAN File System Software Publications CD* is a multilingual CD that includes the Information Center, user publications in PDF format, license information, and the Overview educational module.
- *Master Console Kit CD* contains the master console software and documentation.

Where do I start?

This topic describes some of the more common tasks that you can perform as a SAN File System administrator.

After SAN File System is installed, configured, and the metadata servers are up and running, here are some of the tasks you can perform:

- Set up *administrative security*, including assigning *user roles* to users who will administer SAN File System, to prevent unauthorized access to administrative operations, and set up *client security* to prevent unauthorized access to files in the global namespace.
- Create *filesets*, which are subsets of the entire global namespace. A fileset serves as the unit of workload that can be assigned to a metadata server. After filesets are created, clients can create regular directories and files within those filesets.
- Optionally assign filesets to metadata servers to manually distribute and balance the global namespace workload across all of the servers in your cluster.
- Create storage pools that contain *volumes* that provide specific quality-of-service levels and performance for specific needs, such as storing data for specific applications or business divisions.
- Add volumes, which are LUNs labeled for use by SAN File System, to storage pools so that clients can begin using them to store data.
- Define *policies* and *rules* that determine where newly-created user files are stored.
- Set up *alerts* and *SNMP traps* to ensure timely notification of significant events, such as those that occur in the metadata servers in the cluster, or conditions, such as a fileset or storage pool running out of space.
- Plan and set up a strategy for *backup and recovery* that includes using standard backup tools already available in your environment and SAN File System features, such as *FlashCopy[®] images* (which are point-in-time copies of filesets that can be used for file-based backups) and disaster recovery files (which can be used to recreate system metadata, if necessary).
- View *logs* to obtain additional information about events and conditions.

Note that when SAN File System is installed, the following items exist:

- The *global fileset*, which is the root of the global namespace. All filesets, except nested filesets that are attached to other filesets, are attached to the global fileset using a new directory name.
- A *system storage pool* that is used to store system metadata, which includes information about filesets, storage pools, volumes, and file placement policies.
- A *default storage pool* that can be used to store all files that are not assigned to a specific user storage pool by a rule in the active policy. You must add one or more volumes to this storage pool before you can use the default storage pool. You may also disable the default storage pool.
- A *null policy set* that remains active until an administrator creates and activates a new one. The null policy set assigns all files to the default storage pool.

How can I get quick help with problems?

This topic describes how to get help for SAN File System

SAN File System provides integration with the *master console*, which is a serviceability node for SAN File System and other IBM TotalStorage products. The master console enables an optional *service alert* feature (which notifies IBM service personnel about any problems that occur in your SAN File System environment so that they can respond quickly) and the *remote access* feature (which provides the ability for IBM support personnel who are not located on your premises to assist in diagnosing and repairing failures on a SAN File System engine).

Concepts

This section discusses concepts that will help you understand how SAN File System works. Becoming familiar with the SAN File System components and understanding the concepts in this section enables you to use SAN File System most effectively.

Administrative security

Administrative security protects against unauthorized access to SAN File System administrative operations.

Administrative operations are kept secure with the use of several different levels of authorization access. These access levels are provided through a combination of site preparation steps, use of industry-standard protocols, and configuration of the SAN File System administrative infrastructure.

The steps to prepare your site include the following:

- Setting up your IP network so that all systems that will be used to administer SAN File System by using a Web browser have Internet protocol access to the systems that host the administrative server and metadata servers.
- Ensuring that an LDAP server is installed in your environment. An administrator must define users, their authenticating passwords, and SAN File System administration roles on an LDAP server. Then, the administrator must link users to the SAN File System administration roles.

The SAN File System administrative infrastructure is designed to use the Secure Sockets Layer (SSL) as part of its Web server software-platform base. You log in to the SAN File System console, which is the Web-based user interface, through an SSL-secured login, and your ID is authenticated using the LDAP server.

To use the administrative command-line interface, you must first log in to an engine and be authorized, using the standard UNIX[®] security mechanisms. You can set up security so that SSH-based logins to the cluster are required before the user can use the administrative command-line interface. Then, while using the administrative command-line interface, all access is authorized using the LDAP server.

User authorization

An Lightweight Directory Access Protocol (LDAP) server provides SAN File System with the necessary authentication and authorization for powerful and flexible role-based security. When an administrator issues a request, communication occurs between an LDAP server to authenticate the administrator's user ID and password, and to verify that the user has authority to issue that particular request. Each administrator is assigned a user role that gives that administrator a specific level of access to administrative operations. After authenticating the user, the administrative server interacts with the metadata server on behalf of the user to process the request.

The administrative agent relies on the LDAP server to authenticate and authorize each administrative operation based on your authentication model. This function requires that the LDAP service be readily available when an administrative command is issued. If the LDAP server cannot be reached, the administrative operation fails with an authentication error. Therefore, you must ensure that your LDAP server has high availability.

The administrative agent uses an LDAP cache, so not every operation requires an LDAP query. But when a cache entry expires or is manually cleared, the next operation queries the LDAP server. The administrative agent does not automatically retry LDAP queries; therefore, an LDAP connection failure always results in an authentication error.

User roles

SAN File System provides different levels of user access that you can assign to administrative tasks in your environment. These access levels, or *user roles*, are one way to provide security. The following table describes the SAN File System user roles.

Table 1. SAN File System user roles

Role	Level	Description
Monitor	Basic level of access	Allows you to obtain basic status information about the cluster, display the cluster message log, display the rules in a policy set, and list information regarding SAN File System elements such as storage pools, volumes, and filesets.
Backup	Monitor + backup access	Allows you to perform backup and recovery tasks in addition to all operations available to the Monitor role.
Operator	Backup + additional access	Allows you perform day-to-day operations and tasks requiring frequent modifications, in addition to all operations available to the Backup and Monitor roles.

Table 1. SAN File System user roles (continued)

Administrator	Full access	Provides you with full, unrestricted access to all administrative operations.
---------------	-------------	---

At least one user with Administrator access is required. You can also choose to define other roles as appropriate for your organization.

Administrative server

The *administrative server* processes all requests that are initiated from an administrative interface. Three major components of the administrative infrastructure include IBM Director Agent, a Web server, and the administrative agent.

IBM Director Agent enables remote administration and control of the storage engines.

The Web server is the part of the administrative infrastructure that interacts with the administrative agent and renders the Web pages that make up the SAN File System console. The console is a Web-based user interface, which can be accessed using a Web browser, that has network access to the engines that host the master metadata server in the cluster.

An administrative server interacts with a metadata server through an intermediary service, called the *administrative agent*. The administrative agent is based on the Common Information Model (CIM) standard to process all management requests from the SAN File System console and administrative command-line interface. When you issue a request, the administrative agent checks with the lightweight directory access protocol (LDAP) server to authenticate the user ID and password and to verify whether the user has the authority (is assigned the appropriate role) to issue a particular request. After authenticating the user, the administrative agent interacts with the metadata server on behalf of that user to process the request. It also communicates with the operating system, the Remote Supervisory Adapter II (RSA) card, and administrative agents on other engines when processing requests. This same system of authentication and interaction is also available to third-party CIM clients to manage SAN File System.

To ensure high availability, the administrative server resides on each storage engine. All requests that come from the SAN File System console are always processed by the administrative server that runs on the same engine as the master metadata server. This server is known as the *primary administrative server*. However, requests that are initiated by the administrative command-line interface is processed by the administrative server that is running on the engine that you are logged in to. This can be the primary administrative server or a *secondary administrative server*, which is an administrative server that runs on an engine hosting a subordinate metadata server. All requests are redirected to a secondary administrative server if the primary administrative server is not available.

Alerts and events

An *event* is an occurrence in the metadata server or cluster, such as a change in state from online to offline. Events are recorded as messages in the metadata server logs. You can view these messages using the SAN File System console or by using administrative commands.

An *alert* is a message that can be generated for an event. It warns you about certain conditions, such as a fileset or a storage pool reaching or exceeding its threshold.

A *Simple Network Management Protocol (SNMP) trap* is a notification mechanism to convey the occurrence of an event. You can set configuration parameters that determine whether SNMP trap messages are generated for events. SNMP trap messages notify you about events asynchronously, eliminating the need for you to frequently view messages in the metadata server logs to determine the state of the cluster.

The first configuration parameter determines where SNMP trap messages are sent. An administrator specifies a list of SNMP managers that are the recipients of any SNMP trap messages. The list includes the IP address, port number, version of SNMP, and community string for one or two managers. If no SNMP managers are specified, no SNMP trap messages are sent.

The second parameter specifies which types of event messages also generate SNMP trap messages. Event message types are informational, warning, error, or severe. You can specify any combination of message severities.

In addition, if the call home feature is activated, that feature generates a specific type of SNMP trap whenever a metadata server encounters an event that requires notification to the IBM Support Center. When a metadata server encounters such an event, the following steps are taken by the call-home feature:

1. It sends an SNMP trap Protocol Data Unit (PDU) to the master console
2. It parses and converts the trap into a Simple Mail Transfer Protocol (SMTP) e-mail message.
3. The e-mail message is sent to a known SMTP mail server.
4. The e-mail message is forwarded to the IBM RETAIN[®] system for processing by support personnel.

All of the SNMP-related configuration parameters must be set correctly for the call-home feature to work.

Table 2. Alerts, events, and traps

	Description	Delivery
Alert	Warns of a significant event on a metadata server or cluster. Also informs about condition changes such as a change in state to offline, or storage is approaching capacity	A generated message is sent to the terminal
Event	Identifies an occurrence in the metadata server or cluster	The event is recorded in the metadata server logs.
Trap	Optionally, notifies the administrator of events asynchronously	A generated message is sent to the administrator directly, either locally or remotely

Backup and restore

It is important to have a process for backing up your environment so that you can easily recover from a storage device failure or loss of data. In SAN File System, you must save both the file data and metadata together when you back up the global namespace. These are used to recreate your user data. You must also backup the system metadata, which is used to recreate the SAN File System configuration.

SAN File System does not provide backup and restore functionality; instead, it supports backup tools that are already present in your SAN environment. Depending on the type of failure, you might need to restore a single file, an older version of a file, a directory, a volume, or the entire system. SAN File System supports various options for protecting the system, including:

- Creating FlashCopy images
- Backing up files using third-party backup and restore applications that are already present in your environment, for example IBM Tivoli® Storage Manager (TSM), Legato NetWorker, or VERITAS NetBackup
- Using copy services that exist in the underlying storage device (for example, FlashCopy and Peer-to-Peer Remote Copy (PPRC) functions in the IBM TotalStorage Enterprise Storage Server or IBM TotalStorage SAN Volume Controller
- Saving the cluster configuration (system metadata)

There are two basic methods available for backing up and restoring your data:

- File method
- Volume method

The file-based method saves and restores data at the file level. It uses the FlashCopy function or other third-party backup and restore application in your environment to back up or restore your user data. Use the file-based approach when files have been lost but the overall system remains healthy.

The volume-based method saves and restores data at the device level (that is, a “just-a-bunch-of-bytes” approach). To adopt the volume method, however, the actual copying and restoring of data must be provided as a service by the underlying storage subsystem. Use the volume-based approach when disaster strikes and the system, as well as the FlashCopy images, are unusable.

Tip: Your backup and restore process does not have to be centralized and homogenous, covering the entire SAN, although such a process simplifies the procedure. You can use the volume method even for a fragmented SAN that requires a piecemeal volume copy across two or more storage subsystems. In such a scenario, you would be responsible for manually managing those multiple backup sets as though they were a single backup set.

File-based data backup and restore

In a file-based backup, the smallest unit that you can restore is an individual file. There are two basic methods for backing up files:

- SAN File System FlashCopy, which backs up at the fileset level, but provides the ability to restore parts of the fileset, such as directories, groups of files, or individual files.
- Operating system utilities and vendor-provided backup and recovery tools and applications. These include utilities such as **tar**, **cpio**, **xcopy**, and applications such as Windows Backup, IBM Tivoli Storage Manager, VERITAS NetBackup,

and Legato NetWorker. All these should be able to access the SAN File System global namespace exactly as they would a local drive.

In a situation where files have been lost but the overall system remains healthy, the first line of defense for restoring files is to have previously used the FlashCopy function to create an image of the files. The FlashCopy function is available in SAN File System. The FlashCopy function provides a space-efficient image of the contents of part of the global namespace at a particular moment in time. The FlashCopy image contains read-only copies of the files in a specific fileset as they exist at a specific point in time.

The FlashCopy image is stored in a special subdirectory named `.flashcopy` under the fileset's root attach point. After you create a FlashCopy image of a fileset, you can use standard backup tools on the SAN File System client to back up the files by specifying the path to the FlashCopy image instead of the path to the actual files. This allows users and applications to continue working with the actual files while the backup occurs.

You can use the FlashCopy image to restore the entire fileset or restore a single file.

When using a file-based backup method other than the FlashCopy function, be aware of the associated file metadata backup, which includes all the permissions and extended attributes of the files. This file metadata for Windows-created files can only be backed up completely from a Windows client. Similarly, file metadata for UNIX files can only be backed up completely from an UNIX client. Therefore, if it is important for you to preserve full file-attribute information, create separate filesets by primary allegiance. In other words, have certain filesets that only contain Windows-created files, and other filesets that only contain UNIX-created files. In this way, you can back up the filesets from the appropriate client operating system.

Limitations to file-based backup and restore: Both backup administrators and users on client machines must be aware of limitations that apply when backing up files that are used by both UNIX-based and Windows-based clients.

First, if you request a backup from a UNIX-based client, you must have read permissions on all files and search permissions on all directories (typically a root user). To request a backup from a Windows-based client, you must have read permissions on all files and list-folder-contents permissions on all folders.

Second, to avoid losing security attributes, you must organize the file system in a special way. There are two options:

- The first is to ensure that the root of each fileset contains only directories and folders, and use a naming convention for each that makes it clear to users whether a directory contains files created by an UNIX-based client or a Windows-based client. The top-level directories can contain any combination of files, links, and directories, just as they can within network file system (NFS) and CIFS.
- The second is to choose either UNIX or Windows as the global namespace default, and to require (by convention) that files, directories, and links that are created by non-default clients be created in specially named directories. Users can place these specially-named directories at any level in the file system; however, any file, link, or directory beneath them must also be created by the same type of client.

The special naming conventions are for user and administrator benefit only. SAN File System does not interpret them in any way. In addition, if a directory or folder grants search or “list folder contents” permissions to “other” or “everyone,” SAN File System does not prevent an administrator or any client user from creating a file that violates the convention.

Therefore, administrators and users must be aware of the naming conventions when performing backups, and ensure that backups of files in specially named directories are performed only from the same type of client that created the files. This means that, in an environment that has both UNIX and Windows clients, the backup process must be divided into multiple parts to prevent the loss of security attributes for files. The number of parts could be as few as two, or as many as the number of directories in the global namespace, depending on the capabilities of the native backup utility that a client uses.

Fileset considerations for backup and restore

From the SAN File System client perspective (and therefore from the backup application perspective), a fileset looks exactly like a regular subdirectory. From the metadata server and administrative server perspective, however, this is a fileset that is attached to an arbitrary subdirectory in the global namespace.

When a fileset resides within a directory, backup applications generally create subdirectories automatically when attempting to restore files to a directory that does not exist. In the case of SAN File System, the subdirectory being created might have originally been part of another directory.

Tip: Restore the system metadata backup before restoring any subdirectories and files.

The **mkdrfile** command saves state information that you can later use to recreate this portion of the SAN File System, should there be a disastrous loss of data. Because you can create regular subdirectories only from a client machine, the metadata server cannot recreate directory trees that contain a mix of filesets within subdirectories, and regular subdirectories. To simplify disaster recovery, attach filesets only to the global fileset (root directory) or to each other, but not to regular subdirectories. You can then use the disaster-recovery file created by the **mkdrfile** command to completely restore the top of the global namespace tree before using the client-based backup application to restore the rest of the global namespace.

FlashCopy image considerations for backup and restore

When you make a FlashCopy images, the .flashcopy subdirectory is created as read-only. The client backup application typically backs up the .flashcopy subdirectories along with all other directories or files. At restore time, however, the same backup application attempts to copy the original files back into the same subdirectories and fails. (You would not want the subdirectories there because they would appear to be valid FlashCopy images from the client perspective, when in reality, the metadata needed for the original FlashCopy images would be missing.)

One consequence of this process is that there is no way to restore your original FlashCopy images if you have lost your metadata in a disaster scenario. Only the original files are restored. However, if your backup application has the ability to restore files to a directory other than the files’ original location (that is, to the

grandparent directory two levels above the `./flashcopy/<flashcopypname>` directory), then you have all the ingredients for a highly efficient backup, which leverages the FlashCopy image feature.

Tip: Periodically create FlashCopy images. They are the most efficient method for quickly backing up and restoring files in scenarios where the metadata is still available.

General considerations for backup and restore

These are some backup and restore considerations to keep in mind while planning your backup process:

- UNIX-based clients and Windows-based clients should be used to back up and restore files only within filesets that are dedicated to their respective client type.
- When using the Veritas NetBackup application's standard backup option, do not use the Windows NT[®] File System (NTFS) to ensure proper backups of data. When using Veritas NetBackup to restore data, use the override option, not the default option.
- To simplify disaster recovery, attach filesets only to the global fileset or to other fileset, and not to a directory.
- Backup to a SAN File System disk cannot be done with Windows-based clients using the Veritas Backup Exec 9.0 application; however, SAN File System data can be backed up to a tape device or a local disk, and data can be safely restored back to a SAN File System disk.
- SAN File System does not support the change-journal mechanism for Windows-based clients. Therefore, differential backups using Veritas Backup Exec do not work with SAN File System.
- Veritas Backup Exec's Advance Open File Option package does not work with SAN File System.

System metadata backup and restore

SAN File System manages data and metadata separately. When you back up a file, only the file's data and attributes are backed up. For disaster recovery purposes, you must also back up system metadata (which includes information about fileset attachment points, storage pools, volumes, and policies) separately.

You can create a file that contains a backup copy of system metadata either from the SAN File System console or from the administrative command-line interface using the **mkdrfile** command. The file, which is stored in the `/usr/tank/server/DR` directory on the master metadata server's local disk, contains everything that is required to recreate the system metadata. When needed, you can use the contents of this file (along with normal restore processes for file data) to recreate the state of the cluster.

To restore system metadata, you process the information that is contained in the system metadata backup file using the **bulddrscript** command. This command creates several scripts that you must first review in order to obtain a restore scenario, and then run to recreate the SAN File System configuration. After the system configuration information is restored, you can then restore the user data files from the SAN File System clients.

You should run the **mkdrfile** and **bulddrscript** commands often to ensure that any configuration changes are reflected in the output of these commands. You should store copies of the output of the **mkdrfile** and **bulddrscript** commands in an easily recoverable location on backup media where critical system and application files are kept for backup and restore purposes.

Note: To assist in protecting against the corruption of metadata and other metadata failures, you can check the metadata from the SAN File System console or by using the **startmetadatacheck** command. This command performs a consistency check on the system metadata, and optionally repairs any problems it finds. It allows you to check file metadata for one or more filesets, the system metadata, or both. There is also an option to check only the metadata structure, or to check the metadata structure and its contents.

There are three cases when you might need to perform a consistency check or repair operation:

- As part of a regularly scheduled cycle of preventive maintenance
- In response to an alert that recommends that this operation be performed (extra detail might be supplied that specifies the restore option that you must use in order to salvage the metadata)
- If metadata corruption (or any other SAN File System corruption) is suspected

If the check-metadata operation cannot resolve the problem, you must perform a full restore of SAN File System, beginning with restoring the metadata. It is critical that **mkdrfile** is run in order to recover from such a situation.

Volume-based data backup and restore

The volume-based approach backs up the entire global namespace, at the device level, in a single operation and restores the global namespace as a complete namespace. This approach uses the copy services features that exist in the storage subsystems (for example, the FlashCopy feature of the IBM TotalStorage Enterprise Storage Server).

When performing a volume-based backup, you must be sure to back up the volumes used as volumes in both user storage pools and system storage pool at the same time. All of the volumes in the user storage pools and the system storage pool must be in a static, consistent state, to ensure a static state of the volumes both for the metadata and the user data.

Before performing a volume-based backup, you must quiesce the SAN File System clients and the cluster to ensure a consistent backup. You can also choose to stop the cluster before performing a volume-based backup.

Tip: Use the cluster-transition timestamp to ensure that no unintended cluster transitions occurred during a volume-based backup. The **lscluster** command to displays the cluster statistics.

Advantages to volume-based backup and restore: Using an volume-based backup and restore process has these advantages:

- The backup and restore is performed at the storage subsystem layer, so the storage engines are not involved in the backup process.
- The backup and restore deals with data at the byte level, and has the ability to back up and restore the entire SAN File System global namespace in a single operation.
- The backup and restore of the complete file system (metadata and file data) is performed as it happens at the volume level.

Limitations to volume-based backup and restore: These are some of the limitations to a volume-based backup and restore:

- It is not granular, and does not provide individual file or volume restore capability.
- You must save and restore all the volumes, including those containing metadata and file data. The volumes in the system storage pool and the user storage pools form a consistency group (that is, they must be backed up and restored together).
- The SAN File System clients and the cluster must be quiesced before performing a volume-based backup. In a fully-quiesced system, all file system activity stops and all buffers are flushed to disk. Volume-based backups are guaranteed to get a time-consistent view of the entire system in this state. In a partially-quiesced system, all file system activity stops and only the metadata buffers are flushed to disk. Volume-based backups are guaranteed to get a time-consistent view of the metadata, but not file data because the file buffers are not flushed to disk. Unless your applications can recover from incomplete data writes, your system should be in the fully-quiesced state.

Clients

SAN File System is based on a client-server paradigm. A SAN File System *client* is a computer system that accesses and creates data that is stored in the SAN File System global namespace. A client can act as servers to a broader clientele, providing Network File System (NFS) or Common Internet File System (CIFS) access to the global namespace or hosting applications (for example, database servers or Web-hosting services that use multiple servers).

The *SAN File System Protocol Specification* includes a description of the protocols that are used between a metadata server and clients running on application servers. It is available at www.ibm.com/storage/software/virtualization/sfs.

Clients access metadata (such as a file's location on a storage device) only through the metadata server, and then access data directly from storage devices attached to the SAN. This method of data access eliminates server bottlenecks and provides read and write performance that is comparable to that of file systems built on bus-attached, high-performance storage.

SAN File System supports clients that run several UNIX and Windows operating systems. You must install client software on each client machine. On UNIX-based clients, the software is a virtual file system (VFS). On Windows clients, it is an installable file system (IFS). The VFS and IFS software provide clients with local access to the global namespace on the SAN. A VFS is a subsystem of a UNIX-based client's virtual file system layer, and an IFS is a subsystem of a Windows client's file system.

The VFS or IFS software directs all metadata operations to a metadata server and all data operations to storage devices attached to the SAN. The VFS or IFS software makes the metadata that is visible to a client's operating system, as well as any applications that the client runs, look identical to metadata read from a native, locally-attached file system—that is, it emulates the local file system semantics. In this way, client applications do not need to change their access methods to use SAN File System.

When the global namespace is mounted on a UNIX-based client, it looks like a local file system. When the global namespace is mounted on a Windows client, it appears as another drive letter and looks like an NTFS file system. Therefore, files can be shared between UNIX-based and Windows clients (permissions and suitable applications permitting).

Antivirus software

If more than one SAN File System client is running antivirus software that scans directories and files, shared files only need to be scanned by one SAN File System client. It is unnecessary to scan shared files more than once. When you run antivirus scans from more than one client, schedule the scans to run at different times, to allow better performance of each scan.

Tip: Consider using a single, designated client machine as a backup client to perform all virus scans.

Authentication and authorization

Clients are authenticated using external services such as Active Directory for Windows-based clients and Network Information Services (NIS) for UNIX-based clients. SAN File System does not restrict how authentication is performed, but it does require that all Windows-based clients share a common definition of users and roles, and that all UNIX-based clients share a common domain (definition of users and groups). SAN File System does perform authorization for file access.

All systems that host a metadata server should be part of the shared domain. This makes it easier to perform client-based activities.

Client commands

There is a set of commands (issued from the operating-system command line) that you can use to set up, start and stop the client software and to migrate data. These commands are accessible from each client machine. The client commands are separate from the administrative command-line interface. The client commands allow users to perform operations on the client systems, and the administrative command-line interface allows administrators to administer all aspects of the metadata server.

Host-based clustering

This topic describes the cluster applications that you run on SAN File System clients.

SAN File System works with clients that are in a clustered environment; however SAN File System is independent and not aware of any host-based clustering. Volumes are owned and managed by SAN File System and must not be assigned as resources to the local operating system or cluster manager. Because cluster managers write on the volumes, configure the volumes as raw, unmanaged volumes to each member of the cluster.

With SAN File System, you can use these clustering application on the clients:

- High availability cluster multi-processing (HACMP™) on AIX platforms
- Sun Solaris clustering

Data LUN configuration

You can configure the SAN so that all data LUNs are available to all clients (known as *uniform configuration*) or so that a subset of data LUNs are available to some clients (known as a *nonuniform configuration*). In nonuniform configurations, a client must be able to see all LUNs in any storage pool that has been used or can be chosen by a fileset that is used by that client.

If a client tries to read or write data on a LUN that it cannot see, SAN File System returns an I/O error to that client. If the client performs a file-system operation

that only involves metadata (such as changing a directory, or listing or creating files), SAN File System does not return an I/O error because the operation does not involve the data LUN.

The active policy determines which storage pool is selected when a new file is created. In a nonuniform configuration, the policy must ensure that a newly-created file is allocated to a pool that is accessible to the clients that need the file. When you change the active policy, the new policy must meet the consistency property.

Note: There is no automatic consistency check for a nonuniform configuration; however, when a client identifies itself to a metadata server, the metadata server inspects the client volume list to ensure that no incomplete storage pools are visible to the client. If an incomplete storage pool is visible, the metadata server logs an error in the metadata server log.

File permissions

Newly created filesets are initially attached with a special dedicated user ID and group ID that lock out access to all clients. These are:

UNIX platforms

File permissions 000, userID/groupID 1000000/1000000

Windows platforms

Owner S-1-0-0

For clients to be able to access a fileset, a client must first take ownership of the fileset, by changing the fileset's owner to a valid user that can provide the required access. The take-ownership operation is only performed once for each file system, and can only be done by a privileged client. A *privileged client* is a client machine on which root or Administrator users have the same privileges on the global namespace as they have on other file systems available on their system. A root user logged in to a privileged client is granted full control over directories, files, and other file system objects created by clients.

The concept of *root squashing* means that by default, when a root or Administrator user logs into a client that is not a privileged client, the user's privileges for the global namespace are reduced to that of "Other" in UNIX or "Everyone" in Windows. Therefore, in order to change the ownership and permissions on a fileset, one or more privileged clients must be created. Have at least one privileged client of each client OS type.

In the current release of SAN File System, client files should be separated in filesets for each operating system — that is, a Windows client should create files only within filesets dedicated to Windows files, and an AIX client should create files only within filesets dedicated to AIX. This is referred to as the *primary allegiance* of a fileset — that is, either Windows or UNIX. The different client platforms can, however, share files in a common fileset if the permissions allow. Therefore, it is important to set up your access control list (ACLs) on the clients to accomplish this goal. You should limit your use of cross-platform or heterogeneous file sharing.

To be able to take ownership and change permission on a new fileset, turn off root squashing for the client — that is, enable it as a privileged client to SAN File System.

File sharing

When files are created and accessed from a Windows client, all the security features of Windows are available and enforced. When files are created and accessed from UNIX-based clients, all the security features of UNIX are available and enforced. When files created by a UNIX-based client are accessed by a Windows client, access is controlled using only the semantics and permissions of “other.” Similarly, when files created by a Windows-based client are accessed by a UNIX-based client, access is controlled using only the semantics and permissions of “everyone.”

Restriction: After a directory, folder or file is created from a particular client type (UNIX or Windows), its security settings cannot be changed to another client type.

File sharing in the SAN File System is classified as either homogenous or heterogeneous. File sharing is positioned primarily for homogenous environments. The ability to share files heterogeneously is recommended for read-only—that is, create files on one platform, and provide read-only access on the other platform. Therefore, set up filesets such that they have a *primary allegiance* to a single operating system. This means, for example, that certain filesets have files created in them only by Windows-based clients, and other filesets have files created in them only by UNIX-based clients.

Homogenous file sharing

In a homogenous environment (for example, either all UNIX-based-based or all Windows-based clients), SAN File System provides access and semantics that are customized for the operating system running on the client machines. When files are created and accessed from only Windows-based clients, all the security features of Windows are available and enforced. When files are created and accessed from only UNIX-based clients, all the security features of UNIX are available and enforced.

In homogenous file sharing, the permissions are all one type and are managed within the Windows or UNIX domain as appropriate. Therefore permissions propagate to all the sharing clients. Full support is provided for UNIX and Windows standard file access permissions; however, currently UNIX-extended ACLs are not supported.

In order to facilitate homogenous file sharing, you need UIDs and GIDs (UNIX) or SIDs (Windows) to be consistent in your operating system domains. For example, a uid number 2000 on one UNIX-based system must correspond to the same user with uid 2000 on every other UNIX-based system — and similarly for SIDs (security IDs) with Windows. To facilitate this, a common ID management system is required for each domain (Windows and UNIX), for example, Active Directory for Windows and Network Information Services (NIS) for UNIX, or LDAP, or manual synchronization of ID files. This ensures that permissions granted on one client map directly to other clients.

Heterogeneous file sharing

In a heterogeneous environment (for example, both UNIX-based and Windows-based clients), there is a restricted form of access. When files created on an UNIX-based client are accessed by a Windows-based client, access is controlled using only the semantics and permissions of the “Other” permission bits in UNIX. Similarly, when files created on a Windows-based client are accessed on an

UNIX-based client, access is controlled using only the semantics and permissions of the “Everyone” user group in Windows.

Because the specific permissions do not match exactly between the two operating systems, translation is required. The following table shows the mapping of permissions types between UNIX and Windows. The permissions or ownership can only be changed from the client machine (that is, Windows or UNIX) where the file or directory was created.

Windows permissions	UNIX permissions		
	Read	Write	Execute
Traverse Folder/Execute File			X
List Folder/Read Data	X		
Read Attributes		X	
Read Extended Attributes			
Create Files/Write Data		X	
Create Folders/Append Data		X (parent)	
Write Attributes		X	
Write Extended Attributes		X	
Delete Subfolders and Files		X (parent)	
Delete		X (parent)	
Read Permissions			
Change Permissions			
Take Ownership			
Synchronize		X	

For files created on a UNIX-based system, SAN File System stores the actual uid/gid numbers and shares them across all UNIX-based clients, but they all appear as SID S-1-0-0 on Windows. For files created on Windows, SAN File System stores the actual SID and shares it across Windows clients, but they all appear as 999999/999999 on UNIX. UIDs, GIDs, and SIDs are all mapped by the client to user, group, or owner according to whatever scheme is in use on the client.

Metadata, lock, and data caches

Caching allows a client to achieve low-latency access to both metadata and data. SAN File System supports caching of metadata, locks, and data.

A client caches metadata to perform multiple metadata reads locally. The metadata includes the mapping of logical file system data to physical addresses on storage devices that are attached to the SAN.

A client caches locks to allow the client to grant multiple opens to a file locally without having to contact a metadata server for each operation that requires a lock.

A client caches data for small files to eliminate I/O operations to storage devices attached to the SAN. A client performs all data caching in memory. If there is not enough space in the client’s cache for all of the data in a file, the client reads the data from the shared storage device on which the file is stored. Data access is still fast because the client has direct access to all storage devices attached to the SAN.

Opportunistic locks

Opportunistic locks can be created and used by Windows-based clients. An *opportunistic lock* (or *oplock*) is a lock placed on a file in the global namespace. It allows a Windows-based client to cache data locally, reducing network traffic and improving performance. SAN File System supports level 1, 2, batch, and filter locks.

Direct I/O

Some applications, such as database management systems, use their own cache management systems. For such applications, SAN File System provides a direct I/O mode, which allows these applications to bypass the data cache. In this mode, SAN File System performs direct writes to disk, does not cache data, and allows distributed applications on different computers to write data to the same file at the same time. Using the direct I/O mode makes files act like raw devices. This gives database systems direct control over their I/O operations, while still providing the advantages of SAN File System, such as the FlashCopy feature and file-level backup and restore processing. Applications need to be aware of, and configured for, direct I/O.

UNIX-based clients use existing operating-system interfaces to use direct I/O. That is, you must set the `O_DIRECT` flag to open a file in direct I/O mode. The I/O buffers, offsets and transfer size must be multiples of 512. You receive an `Invalid argument` error if this restriction is not met.

Windows-based clients enforce full, native direct I/O, or *unbuffered I/O*, semantics. You must specify the `FILE_FLAG_NO_BUFFERING` flag to open or create a file in direct I/O mode. When using this flag, your application must meet the following requirements:

- The I/O buffers, offsets and transfer size must be integer multiples of the volume's sector size.
- Buffer addresses for read and write operations must be sector aligned.

You receive a return code of 87 (`ERROR_INVALID_PARAMETER`) if the requirements are not met.

Restriction: You cannot use direct I/O on a file that is being used in cache mode by another process.

Orphaned objects

An *orphaned object* is one that cannot be found in the directory tree. File system objects can become orphaned when the directories that referenced them are damaged. When an object becomes orphaned, it is moved to the lost+found directory. The *lost+found directory* is a special directory that is located at the root of each fileset directory, including the global fileset directory, and each FlashCopy image. It is created automatically when the fileset or FlashCopy image is initialized. It is configured with owner/group 1000000/1000000 and permission 000. It is up to you to change the permissions, as appropriate, depending on whether the fileset is being used by Windows-based clients or UNIX-based clients.

You cannot rename or remove the lost+found directory. You can delete and move objects in the lost+found directory; however, you cannot create new objects in this directory.

Native client file-system security

For UNIX-based clients, SAN File System uses the POSIX definition of three sets of three file mode bits—one set for each user, group, and other. The bits in each group represent read, write, and execute or search permissions. It also uses the SETUID and SETGID bits, and the X/Open-specified restricted deletion mode (also known as “sticky”) bit used for directories. SAN File System supports UNIX commands such as **ls** and **du** when they are run against the global namespace.

If a file created by an UNIX-based client has the read and write bits set for user “other,” all UNIX and Windows users can read and write to the file.

For Windows-based clients, SAN File System uses access control lists (ACLs), which are lists that define permissions for users and groups. An entry in an ACL is called an access control entry (ACE). If a Windows file creates an ACE for user “everyone,” all UNIX and Windows users can access that file.

Privileged clients

SAN File System includes a configurable list of privileged clients. A *privileged client* is a client on which root users in UNIX or users with administrator privileges in Windows are given those same privileges for the SAN File System global namespace. A root user that is logged in to a privileged UNIX-based client is granted full control over directories, files, and other file system objects that are created by UNIX-based clients. A user with administrator privileges who is logged in to a privileged Windows-based client is granted full control over the folders, files, and other file-system objects that are created by Windows-based clients.

If those same users log in to a client that is not a privileged client, their privileges for the global namespace are reduced to those of “everyone” for Windows users or “other” for UNIX users.

UNIX-based clients

A *UNIX-based client* is a SAN File System client that runs a UNIX operating system and has the SAN File System client code installed. In this release, SAN File System supports clients running on these platforms:

- AIX 5.1 (32-bit only)
- AIX 5.2 (32-bit and 64-bit)
- Red Hat Enterprise Linux Advanced Server 3.0, 2.4.21EL kernel
- Sun Solaris 9 (64-bit)

Tip: SAN File System allows AIX clients with up to 8 processors.

The SAN File System client code that is installed on a UNIX-based client is called a Virtual File System (VFS). The VFS is a subsystem of an UNIX-based client’s virtual file system layer. It directs all metadata operations to a metadata server and all data operations to storage devices that are attached to your SAN. The VFS makes the metadata that is visible to the client’s operating system, as well as any applications that run on the client, look identical to metadata read from a native, locally-attached file system.

UNIX-based clients mount the global namespace on their systems. After the global namespace is mounted, you can use it just as you would any other file system to access data and to create, update, and delete files and directories. The following example shows an AIX mount point for SAN File System:

```

root@aix2:/# df
Filesystem 1024-blocks      Free  %Used   Iused  %Iused  Mounted on
/dev          32768      23024    30%   1413     9%    /
/dev/hd1      950272      8096   100%  29103   13%   /usr
SANFS      16728064 16154624 4%    1    1% /sanfs

```

UNIX-based clients use standard UNIX permission semantics (such as read, write, and execute bits, and owner and group IDs) that make the global namespace appear as if it is a local UNIX file system.

Named pipes: With SAN File System, you can create and use named pipes (or FIFO objects) in the global namespace. A *FIFO object* is a standard UNIX feature that is used to communicate and exchange data between processes. It has a directory name and is accessed by a path name. Its file size and a block size are always 0.

When UNIX-based client creates a FIFO object, its file name becomes visible to all other clients, just like any other file. Users and applications on any UNIX-based client can perform standard file-system operations on a FIFO object; however the FIFO implementation is local to the client. In other words, data in a FIFO object is only readable to the same client that wrote the data. Data in FIFO objects is not passed between clients.

Although FIFO objects are visible to Windows clients (subject to file permissions), Windows-based clients cannot create, read from, or write to FIFO objects.

Limitations of UNIX-based clients: The following list describes the limitations of UNIX-based clients:

- UNIX-based clients cannot use user IDs or group IDs 999999 and 1000000 for real users or groups; these are reserved IDs that are used internally by SAN File System.

Tip: To avoid any conflicts with your current use of IDs, the reserved user IDs can be configured at installation time.

- You cannot use multibyte enablement for items referenced in a file-placement policy (such as storage pools, filesets, parts of file names).
- The **mkfs** command is not supported in UNIX.
- All AIX-based clients must reside in the same authentication domain for correct mapping of user and group IDs.
- Clients running on the AIX 5.1 32-bit platform cannot create files that are larger than 1 TB.

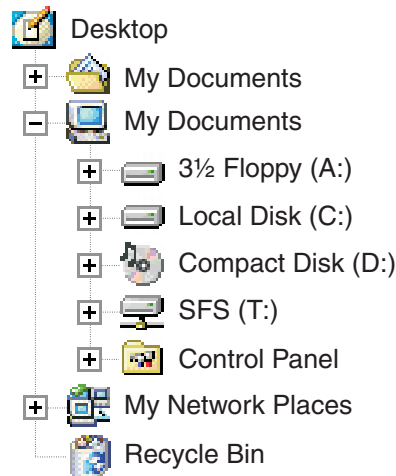
Windows-based clients

A *Windows-based client* is a client that runs a Windows operating system and has the SAN File System client code installed. In this release, SAN File System supports clients that run on these operating systems:

- Windows 2000 Server
- Windows 2000 Advanced Server

The SAN File System client code installed on a Windows-based client is called an Installable File System (IFS). The IFS is a subsystem of the Windows file system. It directs all metadata operations to a metadata server and all data operations to storage devices attached to your storage area network (SAN). An IFS makes the metadata that is visible to a client's operating system, as well as any applications that run on the client, look identical to metadata read from a native, locally attached file system.

Windows clients mount the global namespace on their systems. After the global namespace is mounted, users can use it just as they would any other file system to access data and to create, update, and delete files and directories. The following example shows the My Computer view from a Windows 2000 client. The T: drive (labeled SFS) is the attach point of the SAN File System.



Windows-based clients use a subset of the Windows semantics. The allowed semantics are described to Windows as volume properties, which are visible, for example, as properties of the drive within Windows Explorer. The following volume properties are supported by SAN File System:

- NTFS-like access control lists (which requires all Windows-based clients to share a common Active Directory domain for users and groups)
- Long names and short names (eight-character names with three-character extensions)
- UNICODE-based file names
- Case-sensitive file names

Case sensitivity: SAN File System natively is case-sensitive; however, Windows applications can choose to use case-sensitive or case-insensitive names. This means that case-sensitive applications, such as those making use of Windows support for POSIX interfaces, behave as expected. Native Win32 clients (such as Windows Explorer) get only case-aware semantics.

The case specified at file-creation time is preserved, but in general, file names are case-insensitive. For example, Windows Explorer allows you to create a file named "Hello.c," but an attempt to create "hello.c" in the same folder will fail because the file already exists. If a Windows-based client accesses a folder that contains two files that are created on a UNIX-based client with names that differ only in case, its inability to distinguish between the two files might lead to undesirable results. For this reason, UNIX-based clients should not create case-differentiated files in filesets that will be accessed by Windows-based clients.

Differences between SAN File System and NTFS: SAN File System differs from Microsoft Windows NT File System (NTFS) in its degree of integration into the Windows administrative environment. The differences are as follows:

- Disk management within the Microsoft Management Console shows SAN File System disks as unallocated.

- SAN File System does not support the use of standard Windows write signature on its disks.
- The global namespace cannot be assigned a reserved drive letter.
- Disks used for the global namespace cannot sleep or hibernate.

SAN File System also differs from NTFS in its degree of integration into Windows Explorer and the desktop. The differences are as follows:

- Manual refreshes are required when updates to the SAN File System global namespace are initiated on the metadata server (such as attaching a new fileset).
- You cannot use the recycle bin.
- You cannot use distributed link tracing. This is a technique through which shell shortcuts and OLE links continue to work after the target file is renamed or moved. Distributed link tracking can help a user locate the link sources in case the link source is renamed or moved to another folder on the same or different volume on the same PC, or moved to a folder on any PC in the same domain.
- You cannot use sparse-file APIs or change journaling. This means that SAN File System does not provide efficient support for the indexing services accessible through the Windows "Search for files or folders" function.

File name considerations: File names created on UNIX-based clients using characters that are not valid for the Windows file systems (such as colons, slashes, back slashes, asterisks, question marks, double quotation marks, less than, greater than, and pipe) are transformed into valid short names. Applications can use the short name to gain access to files.

Limitations of Windows-based clients: You cannot use the following features of NTFS when using SAN File System:

- File compression (on either individual files or all files within a folder)
- Encrypted files and directories
- Quotas (however, quotas are provided by SAN File System for filesets)
- Reparse points
- Defragmentation and error-checking tools
- Alternate data streams
- Assigning an access control list (ACL) for the entire drive
- Change journal for file activity
- Scan all files or directories owned by a particular SID (FSCTL_FIND_FILES_BY_SID)
- Security auditing or SACLs
- Windows sparse files

In addition, note these differences:

- Programs that open files using either the 64-bit file ID or the 128-bit object ID (the "FILE_OPEN_BY_FILE_ID" option) will fail. This applies to the NFS server bundled with Microsoft Services for UNIX.
- Symbolic links created on UNIX-based clients are handled specially by SAN File System on Windows-based clients; they appear as regular files with a size of 0, and their contents cannot be accessed or deleted.

Cluster

The SAN File System *cluster* is a set of metadata servers, each running on a separate hardware engine. The metadata servers communicate with each other and with SAN File System clients over your existing IP network. The cluster provides a single point of control for administrative and service operations.

The cluster has one master metadata server, which is initially designated by an administrator, and one or more subordinate metadata servers. The master metadata server maintains the cluster state and is the focal point for most administrative services. The maximum number of metadata servers that SAN File System allows in the cluster is eight.

Note: Although you cannot purchase SAN File System with only one metadata server, you can run a single metadata-server system if all other metadata servers in the cluster fail (for example, if you have only two engines, and one of them fails), or if you want to close all of the metadata servers except one to perform scheduled hardware maintenance.

Cluster workload

Each metadata server in a cluster, including the master metadata server, is assigned a *workload*; a workload is the metadata and locks that a metadata server is responsible for providing metadata to clients whenever they request access. A *fileset* is the unit of workload for metadata servers. Data resides in filesets assigned to metadata servers.

During client setup, a client is given the address of one of the metadata servers for initial contact and metadata server cluster discovery. When the client issues a request to access data, it is automatically directed to the appropriate metadata server to obtain the metadata and locks required to access the data.

SAN File System can automatically balance the workload for you, by dynamically assigning filesets across all metadata servers.

You can also choose to control the workload manually by statically assigning the filesets to specific metadata servers. When assigning static filesets, consider the metadata-activity level of each fileset, and balance the filesets evenly across all metadata servers in the cluster. To ensure that each metadata server handles a share of the entire workload, assign at least one fileset to each metadata server. Creating additional filesets provides you with greater flexibility in assigning and reassigning filesets to achieve optimal results. You can statically assign a fileset to a metadata server or unassign a fileset at any time.

A *spare metadata server* is an idle metadata server that has no statically assigned filesets. You can reserve a spare metadata server for failover to take on the workload of another metadata server that goes offline and preserve the workload balance.

Tip:

- Use all static or all dynamic fileset assignments in the cluster because in mixed environments, filesets could be reassigned to a metadata server that does not have spare capacity to handle the additional workload.
- The global fileset (ROOT) is created during installation time as a static fileset assigned to the master metadata server. You can reassign this fileset to another metadata server or change it to a dynamic fileset.

Communication between metadata servers

The metadata servers communicate with each other for a variety of reasons. For example, they exchange *heartbeats*, which are messages sent periodically from one metadata server to another so that each knows that the other is still active. If a metadata server stops sending heartbeats for a specific period of time (which is set with heartbeat parameters), the other metadata servers form the cluster again without it.

The master metadata server communicates with its subordinate metadata servers to perform many administrative tasks, such as supplying the metadata servers with the current workload map and querying metadata server status. It also contacts them to process requests from administrators to perform tasks such as draining volumes and creating FlashCopy images of filesets.

Subordinate metadata servers initiate contact with the master metadata server for specific tasks, such as acquiring more space or obtaining file placement policy information.

Failover

If a metadata server fails and the metadata-server automatic restart service cannot bring it back into the cluster, or if you manually stop the metadata server, SAN File System automatically and non-disruptively fails over the metadata-server workload by redistributing its filesets and, if necessary, reassigning the master role to other active metadata servers.

SAN File System also detects rogue metadata servers. A *rogue* metadata server is not reachable from the cluster, fails to respond to requests, and might be running or have latent queued I/O. If a rogue metadata server is detected, the cluster first attempts to communicate with the rogue metadata server from disk to have it complete and quiesce all I/O activity that is failing and stops the engine running the rogue metadata server before failing over its workload.

Note:

1. After a failover, review the workload reassignments.
2. Administrative commands that are interrupted by a failover need to be manually restarted against the new master metadata server.

Redistributing filesets

SAN File System attempts to reassign filesets in a useful way across the remaining active metadata servers based on a distribution algorithm. The distribution algorithm first attempts to redistribute the static filesets to a spare, idle metadata server that is set aside for failover. A *spare metadata server* is one that has no static filesets assigned to it. If more than one spare exists, all static filesets assigned the failed metadata server are distributed to a single spare. If a spare metadata server does not exist, the static filesets are treated as dynamic filesets. The dynamic filesets are then distributed in a round-robin fashion among the metadata servers with the fewest number of assigned filesets.

The failover is temporary for static filesets. A *static fileset* is a fileset that you manually assigned to a specific metadata server (using the **mkfileset** or **setfilesetserver** command). These filesets fail back to their statically assigned metadata server when that metadata server rejoins the cluster. *Dynamic filesets*, which are assigned to a metadata server by the system, are not reassigned to their previously assigned metadata server; however, they might be redistributed during failover to rebalance the workload after the static fileset fail back.

Reassigning the master role

When a failure affects the master metadata server, the master role is reassigned to another metadata server according to a quorum algorithm. This algorithm makes use of a quorum disk and a majority voting procedure to assign the master role to a metadata server that is a member of the largest active, mutually-connected group of metadata servers that all have access to the system storage pool.

The quorum algorithm does not take into account the network connectivity between the metadata servers and the clients. If a network partition separates the clients from the metadata server, the chosen master might not be ideal.

Restriction: You cannot specify a preferred master metadata server as the failover target or predict the failover target. Reserve some space capacity for the master role on each metadata server in the cluster. The master role requires only a small amount of processing.

Metadata restart service

The administrative server provides an optional metadata server restart service that monitors the metadata server software and restarts it if necessary. If a metadata server hangs, the Remote Serviceability Adapter (RSA) detects that condition and reboots the engine. If the operating system on the engine stops, SAN File System can reboot the operating system. Then, the metadata server restart service is automatically started, and in turn, restarts the metadata server on the engine. When a metadata server is restarted after a soft failure, it resumes serving the same workload that it was serving before the failure.

Common Information Model

The *Common Information Model (CIM)* is a standard for systems management. It specifies a management protocol as well as the schemas for the various subsystems that it can manage.

The SAN File System console and the administrative command-line interfaces are built around the Common Information Model. These interfaces use protocols that are compliant with CIM standards to communicate with metadata servers in the cluster.

When you issue a request, an administrative agent communicates with an LDAP server through a CIM interface to authenticate your user ID and password and to verify that the user has the authority to issue a particular request. Each administrator is assigned a user role that allows a specific level of access to administrative operations. After authenticating the user ID and password, the administrative agent interacts with the metadata server on behalf of that user to process the request.

The administrative agent, which also functions as a CIM Object Manager (CIMOM) for SAN File System, allows third-party CIM clients to manage SAN File System.

For more information about CIM, refer to the *System Management API Guide and Reference*, GA27-4315.

Components

The following figure illustrates the major components of SAN File System.

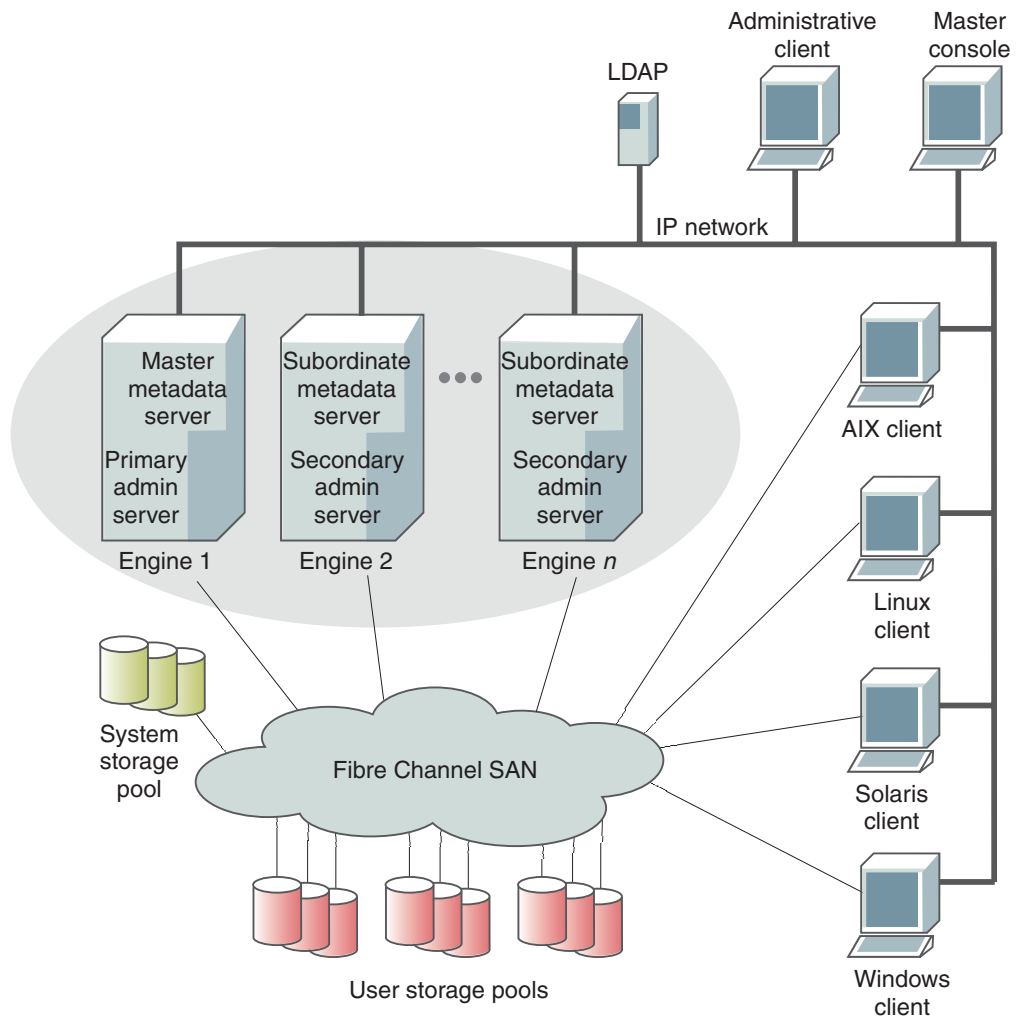


Figure 1. SAN File System components

The metadata servers and clients communicate over an private IP network and access data over a Fibre Channel storage attached network (SAN). SAN File System relies on networking hardware (including an IP network, SAN, network switches, and routers) that already exists in your environment.

The *metadata servers* run on separate physical machines (known as *engines*) and perform metadata, administrative, and storage-management services. The metadata servers are clustered for scalability and availability, and are referred to collectively as the *cluster*. In the cluster, there is one master metadata server and one or more subordinate metadata servers. Additional metadata servers can be added, as required, when the workload grows.

The metadata resides on private storage that is shared among all the metadata servers in the cluster. This storage is known as the *system storage pool*. A storage pool is a collection of SAN File System volumes in the SAN. The system storage pool contains the system metadata (such as system configuration and state information) and file metadata (such as file creation date and permissions). The actual file data is stored on the *user storage pools*, which may be shared among the clients.

The *administrative server* allows SAN File System to be remotely monitored and controlled through a Web-based user interface, called the *SAN File System console*. In addition, the administrative server processes requests issued from the administrative command-line interface, which can also be accessed remotely. The ability to access the SAN File System through these two types of interfaces allows you to administer SAN File System from almost any system with network connectivity. The administrative server uses an *LDAP server* to look up authentication and authorization information about the administrative users. The primary administrative server runs on the same engine as the master metadata server. It receives all requests issued by administrators and also communicates with the administrative servers that run on each additional metadata server in the cluster to perform routine requests.

Computers that are going to share data and have their storage centrally managed are all connected to the SAN. In SAN File System, these computers are known as *clients*. The SAN File System client software enables the clients to access a single, uniform global namespace through a virtual or installable file system. These clients can act as servers to a broader clientele, providing network File System (NFS) or Common Internet File System (CIFS) access to the global namespace or hosting applications (such as database servers or Web-hosting services that use multiple servers).

The *master console* provides serviceability features, including the remote-support interface (or remote access) and service alert (for call home) capabilities. The master console is a required feature for SAN File System that can be shared with other IBM TotalStorage products, such as SAN Volume Controller.

Container

The term *container* is being phased out of SAN File System in favor of the term *fileset*. *Container* still appears in the API, and a few other places. The term *container* means the same as the term *fileset*.

Data migration

Before you can use your existing data in SAN File System, you must first migrate it to the SAN File System global namespace.

Tip: If you are migrating a large amount of data, thoroughly plan the migration in advance, to reduce the risk of error and to minimize downtime to your organization.

You must perform data migration from the client machine. Data from a Windows-based application must be migrated from a Windows-based client. Likewise, data from an AIX application must be migrated from an AIX client.

You can use standard copy commands or utilities that are provided by the client operating system to migrate your data (for example, **cp**, **cpio**, or **tar** commands on AIX; and the **xcopy** command or Explorer on Windows). You can also use backup applications to restore data from the latest backup into the SAN File System as the destination. These methods work best when migrating large numbers of small files.

For migrating large files, SAN File System provides a data migration utility, called **migratedata**, to help migrate your data quickly and efficiently, while preserving the file attributes (such as owner, group, and creation time) of your files. This utility is optimized for bulk data movement of a small number of large files. It includes these features:

- A plan phase to estimate, in advance, the amount of time that the migration operation should take
- A copy phase, when the actual data is copied
- A verify phase, which verifies that the data was successfully migrated
- A transaction-based logging and checkpoint process that allows the migration to be restarted

The **migratedata** utility migrates only the following objects:

- Regular files
- Directories

All other objects in the existing file system are ignored.

The **migratedata** utility attempts to preserve all hard links to regular files. For UNIX-based clients, if SAN File System is set up in such a way that hard links traverse across a fileset, the hard links are replaced with a symbolic link.

You cannot use file-system-based compression (as with NTFS) with SAN File System; however, you can file-based compression (for example, files produced by utilities such as compress on AIX systems and ZIP utilities on systems running the Microsoft Windows operating system). During the migration process, all compressed files are expanded.

Migrating data using the data migration utility is a *disruptive* process. This means that, to guarantee data integrity, you must stop all applications and users from modifying the data being migrated (including database and application servers) until the migration is complete. Only the data being migrated must remain unchanged. To minimize the impact of a migration, you can migrate your data in chunks rather than all at one time.

If you are migrating a database environment, the procedures vary depending on whether your environment is file-system based or contains raw configuration devices. For a file-system-based environment, use the **migratedata** utility to migrate your files, and then reconfigure the database application to reflect the data movement. For raw configuration devices, an unload command to move data out of the raw devices to a temporary holding location, and then perform a load operation to place the data in its location in the global namespace. After the data is loaded into the global namespace, it is file based.

Tip: If you are migrating a Microsoft Exchange database from NTFS to SAN File System, use tools that come with Exchange rather than the **migratedata** utility to ensure that the user configuration data and parameters are also migrated.

Important:

1. The data migration utility does not prevent an application or user from modifying (for example, editing, moving, or deleting) the data being migrated. Make sure that no one modifies this data.
2. The **migratedata** utility writes different parts of a single file in parallel; therefore, there is no guarantee that the file data will be densely allocated. Operating system utilities that copy data sequentially (such as **cp** and **xcopy**) do result in densely allocated file data.
3. On Windows-based clients, migrating from the root of any volume to SAN File System using the **migratedata** utility can set the

destination directory to *system* and hidden. This is as designed because any root volume on NTFS is set to system and hidden.

4. During migration, if there are no other applications running on an AIX-based client, shut down the operating system daemon *syncd* before you start the migration process.

Data-migration considerations

This topic describes considerations for data migration.

Verify the following considerations before starting the data-migration utility:

- SAN File System and clients must be installed and properly configured.
- SAN File System must be set up with the appropriate containers, pools, policies, and security.
- The clients must be able to access the source file systems (for example, directly-attached, network-attached storage (NAS), or storage area network (SAN) disks) and SAN File System during the data-migration process.
- When migrating data from a Windows client, you must create the destination directory in the SAN File System file system and verify that the security attributes of the destination directory match that of the source directory. Otherwise, the verification phase of data migration will fail and the migrated data will have incorrect permissions.
- All applications that modify the data being migrated (including database and application servers) must be stopped until the migration completes to guarantee data integrity.
- Twice the space of the data is available for migration. Note that the data-migration utility does not verify whether there is enough space in the system pools where data is being migrated.
- Compressed files are expanded during data migration. Sufficient space must be available in the SAN File System to store the expanded files. Refer to the documentation for your operating system to determine the compression ratio and estimate the amount of space required.
- Sparse files become dense, or full, files during data migration. Sufficient space must be available in the SAN File System to store the dense files.
- To invoke the **migratedata** command, you must either supply the full path or update the PATH environment variable to include the migration directory.
For clients based on UNIX, this is done in your shell profile (for example, export PATH=\$PATH:/usr/tank/migration/bin).
For clients for Windows, edit the PATH environment variable (for example, from the Control Panel, double-click **System**, and then click the **Advanced** tab and **Environment Variables**. In the **User variables** group, click the PATH variable and then click **Edit**. At the end of the text in the **Variable Value** field, type: **c:\Program Files\IBM\Storage Tank\Migration**).
- You must have superuser privileges (for UNIX-based clients) or administrator privileges (for Windows clients) to migrate data.

Engines

Within SAN File System, the hardware on which a metadata server and an administrative server run is called a storage *engine*. SAN File System supports from two to eight engines.

SAN File System is intended to run with a minimum of engines. However, you can run a single-engine system if:

- All of the other engines fail (for example, if you have only two engines, and one of them fails)
- You want to bring down all of the engines except one before performing scheduled maintenance.
- One engine hosts a spare metadata server.

You can use the SAN File System console or administrative command-line interface to monitor and control the engines from any computer with a network connection to the cluster.

Filesets

In most file systems, a typical file hierarchy is represented as a series of folders or directories that form a tree-like structure. Each folder or directory could contain many other folders or directories, file objects, or other file-system objects, such as symbolic links or hard links. Every file system object has a name associated with it, and it is represented in the namespace as a node of the tree.

SAN File System introduces a new file system object, called a *fileset*. A fileset can be viewed as a portion of the tree-structured hierarchy (or global namespace). It is created to divide the global namespace into a logical, organized structure. Filesets attach to other directories in the hierarchy, ultimately attaching through the hierarchy to the root of the SAN File System cluster mount point. The collection of filesets and their content in SAN File System along with the file system root combine to form the global namespace. Fileset boundaries are not visible to the clients; only the administrator of SAN File System can see them.

From a client's perspective, a fileset appears as a regular directory or folder within which the clients can create their own regular directories and files. Clients cannot delete or rename the directories that represent filesets.

In addition to organizing the overall structure of the global namespace, SAN File System also uses filesets for these purposes:

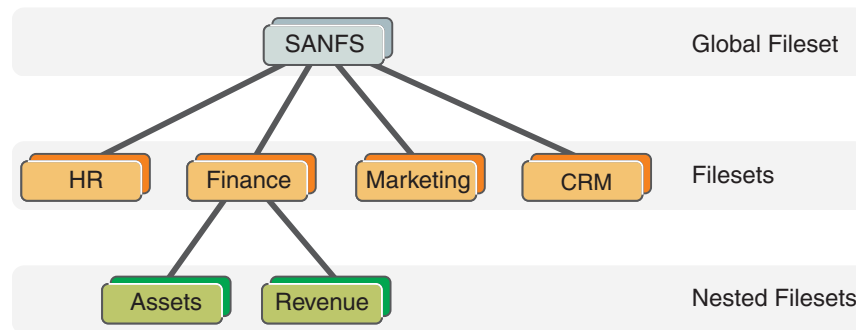
- Represent a unit of workload for the metadata servers
- Provide a level of granularity for data replication (using FlashCopy images)
- Control the amount of space used by the clients (through hard and soft quotas)

A fileset has the following properties:

- A fileset name.
- A directory path leading to the directory within which the fileset is attached. The directory path for the global fileset is the same as the cluster name, *sanfs*.
- A directory name that the fileset is given at the end of the directory path.
- A hard or soft quota.

The root of the global namespace is the *global fileset*. The name of the global fileset is always ROOT. The directory path of the global fileset is specified when you set up the global namespace and is the same as the cluster name *sanfs*.

When you create a fileset, you attach it to a specific location in the global namespace, either to the global fileset or to another fileset. When a fileset is attached to another fileset, it is called a *nested fileset*.



You can detach a fileset and reattach it at the same location or a different location. If a fileset is reattached at a different location, all the files contained in the fileset are rooted to the new location without any further operations. Before a fileset can be detached, any nested filesets must be detached first.

Filesets and clients

From a client perspective, a fileset appears to be a regular directory. Users and applications running on the clients can create objects, such as directories and files, within the fileset.

A fileset must be attached to the global namespace before it is available for use by clients.

Users cannot create hard links across fileset boundaries. In addition, a user cannot rename, move, or delete a directory that is the root of a fileset. If a user attempts to perform any of these operations, global namespace issues an error message.

Filesets and metadata servers

When creating a fileset, you may statically assign it to a specific metadata server or SAN File System can dynamically assign it to a metadata server for you. That metadata server is then responsible for providing metadata and locks to clients when they request access to files that reside in that fileset. The fileset-to-metadata server assignment is automatically communicated between clients and metadata servers. The client transparently discovers which metadata server to use when accessing files in a fileset. Each metadata server should be assigned to manage one or more filesets. If a metadata server is not managing any filesets, it is considered to be in standby mode. You can have an idle, or nearly idle, metadata server available to provide failover, if desired.

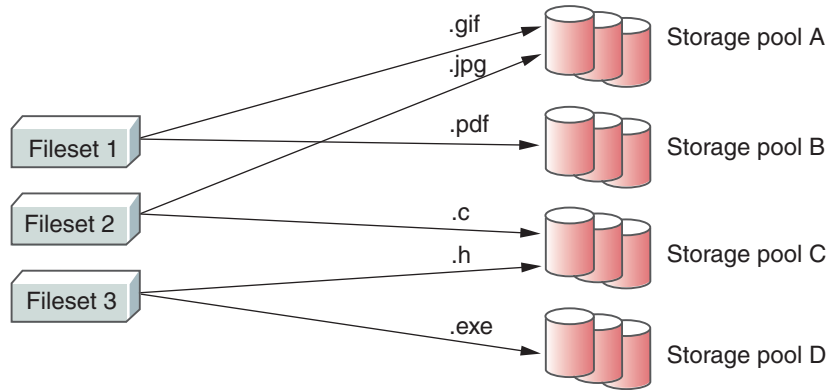
You should create at least one fileset for each metadata server in the cluster. However, creating more filesets gives you greater flexibility in distributing filesets among metadata servers in order to maintain availability and to balance the workload.

Tip: You can assign a nested fileset to a different metadata server than the one to which its parent fileset is assigned.

You can reassign a fileset to another metadata server, for example, to balance the workload. While filesets are being reassigned, they are temporarily unavailable to clients. After the reassignment, the clients can continue transparently and will automatically recognize the new metadata server hosting the fileset.

Filesets and storage pools

Filesets are not specifically related to storage pools, although each file in a fileset physically resides in blocks in a storage pool. This relationship is many-to-many; each file in the fileset can be stored in a different user storage pool. A storage pool can contain files from many filesets. However, all of the data for a particular file is wholly contained within one storage pool. The following illustration shows an example of the relationship between filesets and storage pools.



When you create file placement policies, you can specify that all files created in a particular fileset are to be stored in a specific storage pool.

Fileset considerations

You can create filesets based on conditions in your environment (for example, workflow patterns, security, or backup considerations, all the files used by a specific application, or files associated with a specific application or client). Filesets are used not only for managing the storage space used, but also for creating FlashCopy images. Correctly defined filesets mean that you can take a FlashCopy image for all the files in a fileset together in a single operation, providing a consistent image for all of those files. The global namespace is partitioned into filesets that match the data-management model of the enterprise. Filesets can also be used as a criteria when placing individual files in global namespace.

Tip: When you are creating filesets, consider the overall I/O loads on the cluster. Because each fileset is assigned to one (and only one) metadata server, you need to balance the load across all metadata servers in the cluster by assigning filesets appropriately.

Separate filesets by their *primary allegiance* of the operating system to facilitate file sharing. Separating filesets also facilitates file-based backup methods (for example, utilities, such as tar, and Windows backup applications such as VERITAS NetBackup or IBM Tivoli Storage Manager); full metadata attributes of Windows files can be backed up from a Windows backup client only and full metadata attributes of UNIX files can be backed up from an UNIX backup client only.

There are no known limitations to the number of filesets that you can create; however, when the number of filesets is greater than one thousand, response time will increase when you issue fileset commands.

Fileset permissions

When you create and attach a new fileset to the global namespace, the fileset is owned by user *Anonymous*. A UNIX root user or a Windows administrator user must change the ownership and permissions of the fileset before the fileset is usable. (You must do this for the FlashCopy directory and the lost+found directory under the fileset root.) You need to make these changes only once in the lifetime of

a fileset. The changed permissions are persistent across metadata server restarts and whenever the fileset is detached or attached.

Unlike the requirement for the global fileset, a UNIX or Windows user can own a fileset exclusively. The fileset is not required to have write permissions for both UNIX and Windows domains.

Tip: If you change the permissions of a fileset after you create a FlashCopy image and then revert back to that FlashCopy image, the permissions also revert to their settings at the time when the FlashCopy image was taken.

Fileset quotas

When creating a fileset, you can specify a maximum size for the fileset, called a *quota*, and specify whether SAN File System should generate an alert if the size of the fileset reaches or exceeds a specified percentage of the maximum size, called a *threshold*. For example, if the quota on the fileset is set to 100 GB, and the threshold is 80%, an alert will be generated when the fileset contains 80 GB of data. (Note that the quota is based on space allocated to the fileset, not the data it contains.)

The action taken when the fileset reaches its quota size depends on whether the quota is defined as hard or soft. If a hard quota is used, once the threshold is reached, new client requests to add more space to the fileset (by creating or extending files) are denied. If a soft quota is used, which is the default, more space can be allocated but alerts continue to be sent. Once the amount of physical storage available to global fileset is exceeded, no more space can be used. The quota limit, threshold and quota type can be set individually for each fileset.

Note:

- The space used by a fileset includes the space used by FlashCopy images. It does not include the space used by any filesets nested within it.
- The metadata servers compute and track hard quota limits for filesets in multiples of the partition size. If a hard quota is not set as a multiple of the partition size, quota violation errors appear in the log file even though the size of the fileset has not reached the specified limit. To avoid this problem, specify hard quota limits as multiples of the partition size (for example, if the partition size is 16 MB, set the quota to multiples of 16).

Nested fileset considerations

Consider the following circumstances when creating nested filesets:

- You cannot access a nested fileset if the metadata server that is hosting the parent fileset is unavailable. In other words, if the metadata server hosting a fileset is offline, any nested filesets, even if hosted by a different metadata server, would also be offline.
- A FlashCopy image is created at the individual fileset level and does not include any nested filesets. You cannot make a FlashCopy image of a fileset and any nested filesets in a single operation. This can be of concern if you are required to have a consistent image of a fileset and its nested filesets. Making FlashCopy images in multiple operations could lead to ordering or consistency issues.
- To detach a fileset, you must first detach all of its nested filesets.
- It is not possible to revert to a FlashCopy image when nested filesets exist within the fileset. You must manually detach the nested filesets before reverting to the image. You can reattach the nested filesets after the fileset is reverted.

- When creating nested filesets, attach them only directly to other filesets. Do not attach filesets to client-created directories because a large-scale restore will be more complex.

FlashCopy images

The IBM TotalStorage SAN File System has a FlashCopy function, which creates an instantaneous copy or image of a fileset. The created image is a read-only, space-efficient image of the contents of a SAN File System fileset at the time that it was taken. You can use standard backup applications or utilities on SAN File System clients to back up the contents of FlashCopy images, rather than the actual fileset. Backing up the image avoids any issues with open files that might cause problems when backing up live data.

FlashCopy images are file-based, so the SAN File System clients can see all of the files and directories in the image. The clients can use this image for quick restore of parts of the fileset if required, by simply copying the required files and folders back to the actual fileset. You can also quickly revert the entire fileset from a FlashCopy image.

Backing up and reverting files using FlashCopy images

When performing a backup using standard backup tools available in your environment, a client can specify the path to the FlashCopy image instead of the path to the actual files and continue working with the files while the backup occurs. This procedure produces a consistent backup of the files in the fileset.

Although creating FlashCopy images is *not* a replacement for creating backups of your files to protect your data, in some cases, a user can choose to use a FlashCopy image to revert a file or a set of files to a specific point in time. For example, if a user accidentally deletes a file, restoring it by copying it from a FlashCopy image to another directory instead of restoring it from a backup copy can be faster.

When you restore files from a backup taken from a FlashCopy image, you cannot restore the files to the same location – all FlashCopy image directories are read-only directories. You must restore the files to the directory where the original files resided or to another directory.

For ease of management, create FlashCopy images of all filesets at the same point in time, and use a common naming convention to indicate that they represent a set.

Copy on write

Immediately after the FlashCopy operation, the original fileset files (the source data) and the FlashCopy images (the copy data) of the files in the fileset share the same data blocks; that is, nothing is actually copied, which makes the operation space efficient.

As soon as any updates are made to the actual fileset contents (for example, a client adds or deletes files, or updates contents of files), the fileset is updated by an operation called *copy on write*; in which only the changed blocks in the fileset are written to a new location on disk. The FlashCopy image continues to point to the old blocks, whereas the actual fileset will be updated over time to point to the new blocks.

For example, when you create a FlashCopy image, the image is a set of pointers back to the original data in the fileset.

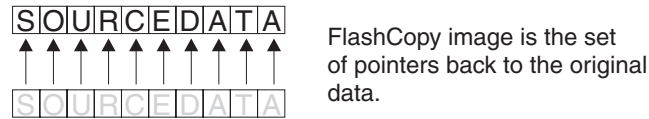


Figure 2. FlashCopy image

If two blocks were changed (S and E), one block was deleted (T) and a new block was written (P) in the actual fileset, the new blocks would be written, and the FlashCopy image continues to point to the original blocks, preserving the point-in-time copy.

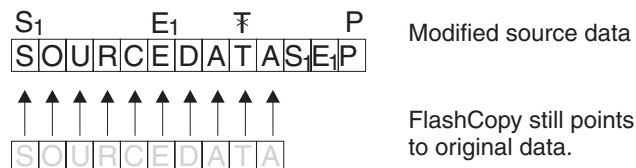


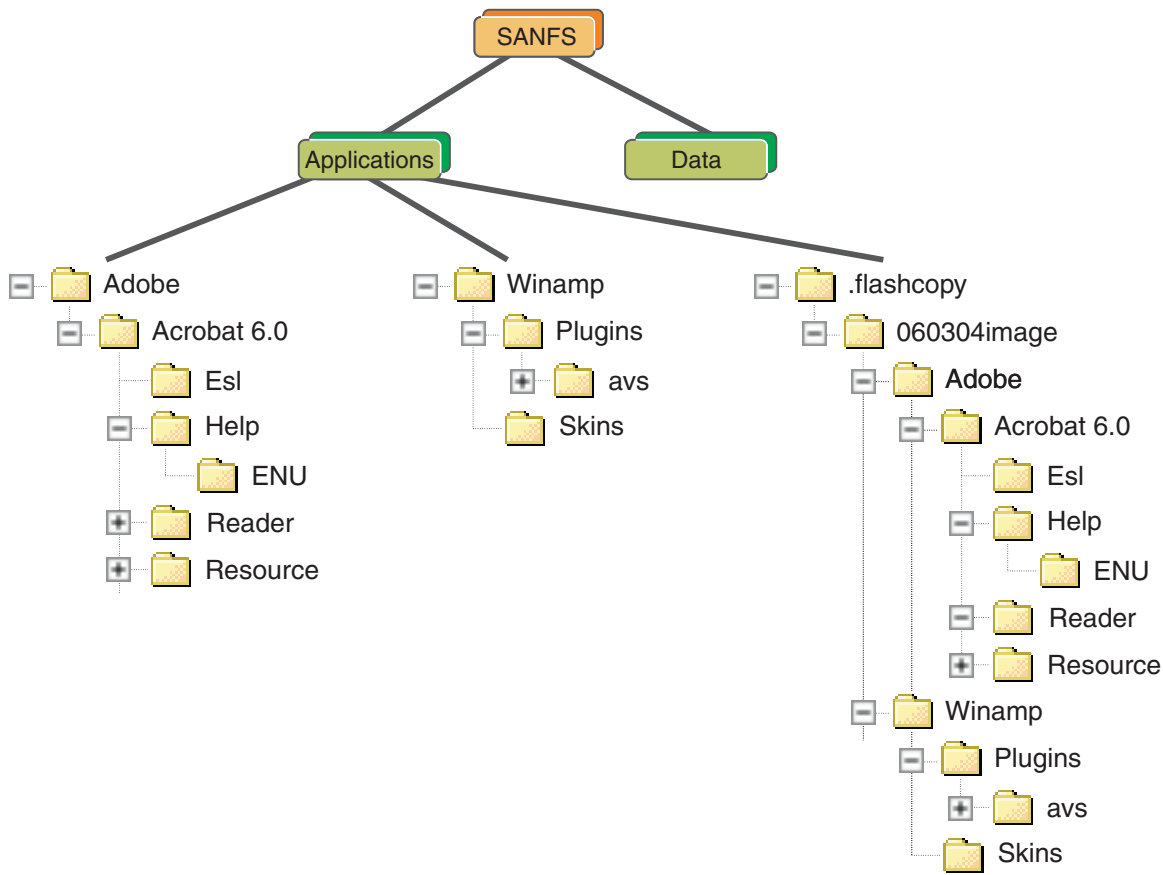
Figure 3. Copy on write

Therefore, any access to the FlashCopy image accesses the data blocks as they existed when the FlashCopy image was created, and any access to the fileset itself accesses the new data blocks.

Creating FlashCopy images

FlashCopy image is an image of an entire fileset as it exists at a specific instant. When you create a FlashCopy image, you specify the fileset to be copied. The FlashCopy image operation is performed individually for each fileset. While the FlashCopy image is being created, all data remains online and available to users and applications. The space used to keep the FlashCopy image is included in its overall fileset space; however, a space-efficient algorithm is used to minimize the space requirement. The FlashCopy image does not include any nested filesets within it. Also, you can create incremental FlashCopy images to be used as the basis for incremental backups. You can create and maintain a maximum of 32 FlashCopy images of any fileset.

The following figure shows how a FlashCopy image is viewable on a Windows client. In this case, a FlashCopy image was made of the Applications fileset and created in the directory 060304image. The fileset has two top-level directories, Adobe and Winamp. After the FlashCopy image is made, a subdirectory called 060304image appears in the special directory, .flashcopy (which is hidden by default), under the root of the fileset. This directory contains the same folders as the original fileset (Adobe, Winamp, and all the files and folder structure underneath). It is captured at the time the image was taken. Therefore, clients have file-level access to these images, and can access older versions of files, or make copies of individual files for the original fileset, if required.



When creating a FlashCopy image for a fileset, you can indicate whether the oldest image should be deleted if creating a new one causes the maximum number of images to be exceeded. Once a FlashCopy image is created, its name cannot be changed.

The actual files in a fileset and the FlashCopy images of the files in the fileset share the same file data blocks until a client makes changes to the files. When a client makes a change to a file, such as adding or deleting data, the client performs an operation called *copy on write*, in which the client writes the changed blocks to a new location on disk. At this point, the FlashCopy image points to the old blocks, and the actual file points to the blocks with the new data. Therefore, any access to the FlashCopy image produces the data blocks as they existed when the FlashCopy image was created, and any access to the actual file accesses the new data blocks.

FlashCopy image considerations

These are some basic considerations regarding FlashCopy:

- A FlashCopy image is simply an image of an entire fileset as it exists at a specific point in time.
- While a FlashCopy image is being created, all data remains online and available to users and applications.
- The FlashCopy image operation is performed individually for each fileset — that is, you can create only one FlashCopy image at a time.
- FlashCopy images are full images — you cannot create incremental FlashCopy images.
- Each fileset can have up to 32 read-only FlashCopy images.
- Once a FlashCopy image is created, its name cannot be changed.

- You can use a FlashCopy image for backing up files, instead of the original source data. This guarantees a consistent image of the files because the files in a FlashCopy image are read-only.
- Clients have file-level access to FlashCopy images, in order to access older versions of files, or to copy individual files back to the real fileset if required.
- FlashCopy images for each fileset are stored in a special subdirectory called `.flashcopy` under the fileset's attachment point. The `.flashcopy` directory is a hidden directory, so by default, it does not appear in Windows Explorer in a SAN File System client.

Disk space used by FlashCopy images

FlashCopy images consume space on the same volumes as the original fileset. Because FlashCopy uses a space-efficient method to make the image, the amount of space that is used by FlashCopy images is not possible to predict. If all blocks in the fileset are changed, the image takes up the same amount of space currently occupied by the non-FlashCopy objects within the fileset. If nothing in the fileset changes, the FlashCopy images takes up virtually no space (just pointers to the real fileset data). It is not possible to determine how much space is being occupied by a particular FlashCopy image at any particular time.

Therefore, when planning space requirements, include space for FlashCopy images. The amount of space you need to plan for flashcopy images correlates to the amount of changes you make to files with flashcopy images. Carefully monitor the user-storage-pool space threshold. Be aware that the space used by FlashCopy images count toward the fileset's quota.

Global namespace

The *global namespace* is the key to SAN File System. It allows common access to all files and directories to all SAN File System clients, and ensures that all SAN File System clients have consistent access and a consistent view of the data and files managed by SAN File System. Having common access to all files reduces the need to store and manage duplicate copies of data and simplifies the backup process. Security mechanisms, such as permissions and access control lists (ACLs), restrict visibility of files and directories.

Client access to the global namespace

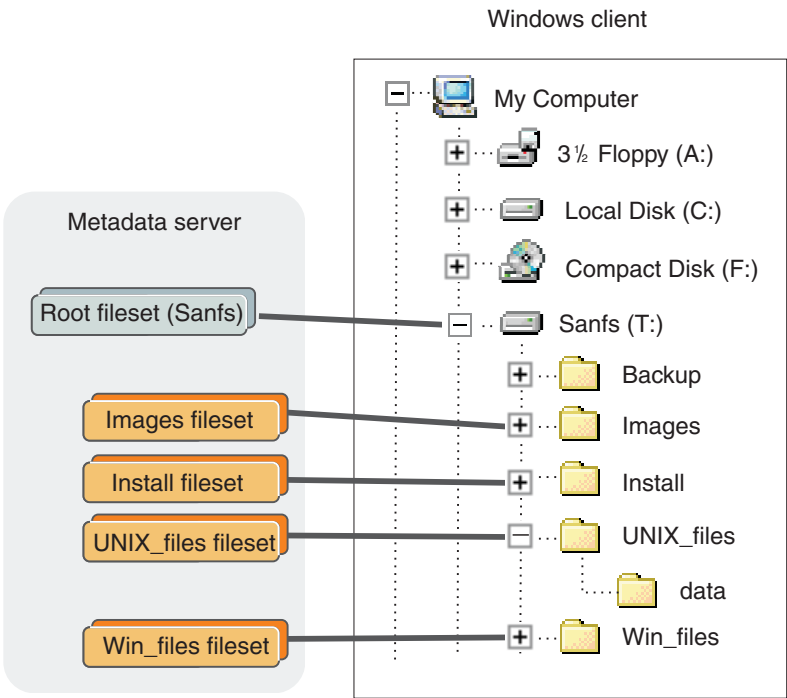
SAN File System clients mount the global namespace on their systems to access the filesets. After the global namespace is mounted on a client, users and applications can use it just as they do any other file system in order to access data and to create, update, and delete directories and files.

From a client's perspective, the global namespace appears as a normal directory. On a UNIX-based client, the global namespace looks like a mounted file system. On a Windows client, it appears as another drive letter and looks like any other NTFS file system. Basically, the global namespace looks and acts like any other file system on a client's system.

Note: A client cannot move, rename or delete a fileset, and cannot create hard links across fileset boundaries.

The following figure illustrates the appearance of the fileset from the metadata server and client perspectives. There are five filesets shown: the root, Images, Install, Unix_files, and Win_files. Some of these filesets have subdirectories (for example, the folder Backup is a subdirectory on the root filesystem, and the fileset

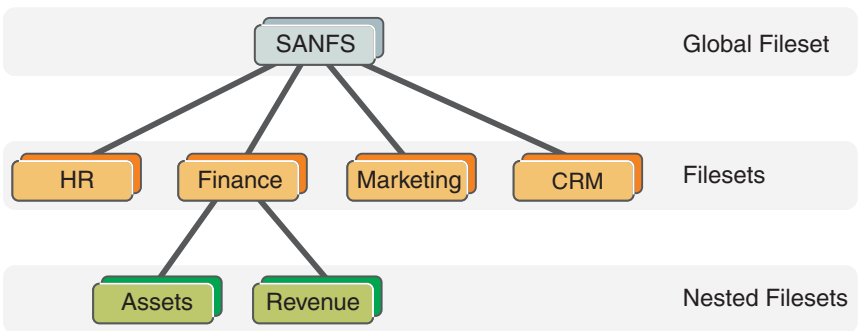
Unix_files, has a subdirectory named data). The client, however, cannot tell which folders are filesets; they appear all as regular directories.



Global namespace structure

The global namespace is organized into filesets. Each fileset is available to the global namespace at its attachment point. You are responsible for creating filesets and attaching them to directories in the global namespace. This can be done at multiple levels. An attach point appears to a SAN File System client as a directory in which the client can create files and directories (permissions permitting).

The following figure shows a sample global namespace. In this sample, the global fileset is attached to the root level in the namespace hierarchy (sanfs), and the filesets (HR, Finance, Marketing, and CRM) are attached to the global fileset, and the nested filesets (Assets and Revenue) are attached to the Finance fileset. By defining the path of a fileset’s attach point, you also automatically define its nesting level in relationship to the other filesets.



Shared access to the global namespace

A *homogeneous environment* is one in which all clients run the same operating system. In a homogeneous environment, SAN File System provides access and

semantics that are customized for the operating system that is running on the clients. For example, when files are created and accessed from only Windows clients, all the security features of Windows are available and enforced. When files are created and accessed from only UNIX clients, all the security features of UNIX are available and enforced.

A *heterogeneous environment* is one in which clients run more than one type of operating system. In a heterogeneous environment, there is a restricted form of access. For example, when files created on an UNIX client are accessed by a Windows client, access is controlled using only the semantics and permissions of the “other” permission bits in UNIX. Similarly, when files created on a Windows client are accessed on an AIX client, access is controlled using only the semantics and permissions of the “everyone” group in Windows.

Locks and leases

SAN File System uses locks and leases to ensure the consistency and integrity of data in the SAN File System global namespace. The internal locks discussed are in addition to the locks provided with the native file systems, such as flock() in UNIX.

A *lock* is a mechanism that restricts access to data and metadata. The SAN File System protocol provides locks that enable file sharing among SAN File System clients, and, when necessary, provides locks that allow clients to have exclusive access to files. It uses distributed data locks for cache consistency and file access locks to synchronize multiple, concurrent open instances of the same file. You can use locking semantics that correspond to open modes that are native to Windows and UNIX operating systems. You can also use byte-range locks.

When a client or server fails, SAN File System uses a lease-based safety protocol to ensure data consistency and to protect the structural integrity of the global namespace.

A client obtains a *lease* from a metadata server as soon as it makes contact with that server. A lease is valid for a period of time that is set by an administrator using a metadata server configuration parameter. When a client obtains a lock from a server, that lock is guaranteed to be valid by the server only as long as the client has a valid lease with the server. The server renews a client’s lease each time the client contacts the server.

If a client does not contact the server within the specified lease period (for example, because of a temporary network failure), the server can revoke the client’s locks. If other clients request locks on the same data, the server revokes the first client’s locks and grants new locks to the other clients. If no such requests are made, when the client contacts the server again, it can renew its lease and reassert any locks (get its old locks back) that protect modified but uncommitted data in the client’s cache, thus preventing data loss.

A client can also lose its lease because of a server failure. However, when the metadata server is restarted, the client can renew its lease and reassert its locks. A metadata server provides a grace period for lock reassertion to allow clients to reassert their locks before allowing other clients to obtain new ones. Reasserting the locks preserves the first client’s cache. Clients cannot access any new data until the grace period has ended.

Logs and traces

SAN File System provides various logging and tracing mechanisms for use with the metadata server and clients. Logging messages provide a trail of routine system activities, operations that occur in normal day-to-day use of the product; as such, they are of interest to system administrators, as well as to IBM Support Center and trained service personnel. The contents of the logs, together with the observed symptoms, are used as a basis to begin isolating problems.

You can view the administrative, audit, cluster, and security logs from the SAN File System console or by using the **catlog** command; to clear the cluster and audit logs use the **clearlog** command. The SAN File System console also allows you to filter these logs based on severity and date.

Messages in the SAN File System console log views are linked to the Information Center. Clicking the message ID displays the Information Center with a description of the message and recommended actions to resolve the problem.

Administrative log

An *administrative log* contains entries for routine activity and error conditions that are generated by the administrative servers. A separate log is maintained for each administrative server. If you access these logs through the master metadata server (either from the SAN File System console or the administrative command-line interface, you see a consolidated view of the logs for each administrative servers in the cluster, ordered by date and time. If you access these logs through a subordinate metadata server (from the administrative command-line interface), you see only the logs for the administrative server running on the same engine as that particular metadata server.

Audit log

The *audit log* contains administrative audit messages, which are generated in response to operations performed by the SAN File System administrative server. The log does not capture every administrative operation, but records all commands that modify metadata or cluster configuration, or are significant operations, including commands that would have made such a change but failed to do so. The log also keeps a record of the user ID issuing the command, along with the time stamp and completion status of the requested operation. The log does not keep a record of simple query operations; such operations do not alter metadata, and because they are likely to be more numerous than those that do, their presence could easily overwhelm logging and interpretation of more meaningful operations.

The audit log is located on the engine that is hosting the master metadata server. When the master metadata server switches to another engine, the audit log on the original engine merges with the audit log on the new engine.

You can use information in the audit log to help convert requests made from the SAN File System console into equivalent command line interface instructions or to perform troubleshooting in the case of a failure within SAN File System.

Client logs and traces

Use the logging and tracing functions that are provided on each SAN File System client to help diagnose client-related problems. There is also a client dump facility to help with data collection in the event that IBM Support personnel requests this information.

Windows client logs, traces, and dumps

For Windows users, client log messages are written to the standard event log. To view the log, click **Start -> Programs -> Administrative Tools -> Event Viewer**.

Windows client tracing is written in the file C:\Program Files\IBM\Storage Tank\Client\log\sanfs.log for Windows. You must first enable tracing, using **stlog** command. You can view the trace log using a standard text editor.

You can configure Windows to generate a dump file if the Windows SAN File System client ends abnormally. By default, the file name is C:\WINNT\memory.dmp.

AIX client logs, traces, and dumps

In AIX, use the **stfsdebug** command and the **syslog** utility to enable tracing and logging on an AIX client. This client generates both log and trace messages, which are routed through the **syslog** utility on the AIX operating system. The **syslog** utility captures log and trace output from the kernel, as well as other operating system services. By default, the **syslog** utility discards all kernel output. However, you can configure the **syslog** utility to specify a destination for the messages by modifying the `/etc/syslog.conf` file. You use the **stfsdebug** command to enable tracing for an AIX client. In addition, you can specify which components (called classes) are traced, as well as the level of detail to include. You can also use **stfsdebug** to query the current status of all trace classes. Trace output enabled **stfsdebug** is sent to the syslog facility.

You can initiate a kernel dump of an AIX system if it is still running but no longer responding to commands. Initiate a kernel dump using the **sysdumpstart** command from a system prompt using telnet or Secure Shell (SSH) tools; if you cannot establish remote access to the client, press the system reset button.

Linux client traces

To turn on tracing for the Linux client:

1. Verify that there is an entry in `/etc/syslog.conf` file for `kern.debug` (for example, `kern.debug /var/log/kernmsg`).
2. After the client module is loaded, edit the proc entries:
 - a. Edit the `"/proc/fs/stfs/debug"` entry turn on tracing for the needed components (for example, change OFF to ON).
 - b. Edit the `"/proc/fs/stfs/csmdebuglevel"` entry to set the CSM tracing level.

The trace output is placed in the location specified in the `/etc/syslog.conf` file (for example, `/var/log/kernmsg`).

Solaris client logs, traces, and dumps

You use the **sanfs_ctl** utility to enable tracing on a Solaris client. Logging and tracing output is stored in the standard system log in file `/var/adm/messages`.

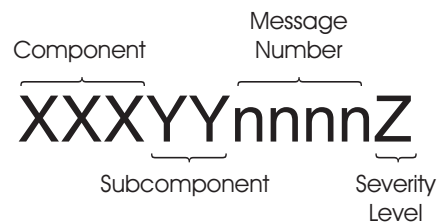
The Solaris operating system automatically creates a disk copy of the physical memory when a system crash occurs. This disk copy is known as a crash dump. The crash dump information is saved in a set of files named `unix.x` and `vmcore.x`, where *x* is an integer that identifies the dump. You can configure the directory in which these files are saved and can set other dump parameters using the **dumpadm** utility.

Event log

The event logs contains a subset of the entries in the metadata server logs. The entries that appear in the event logs are those events, such as changes in server state, that have been configured as alerts. The event log is not stored in a separate physical file, but is generated from entries in the metadata server logs or cluster log.

Message IDs

The format of the message IDs helps you determine the type of error. The format of the message IDs is shown in the following figure. The *XXXX*, *YY*, and *Z* fields are alphabetic, and the *nnnn* field is a 4-digit number. For example, HSTAD001I would be an informational Basic Administration Message from the SAN File System Administration Service.



Refer to the “Message conventions” topic in the reference section for a detailed list of values for the component and subcomponent fields.

The severity can be one of these values:

I	Informational
W	Warning
E	Error
S	Severe

Metadata server logs

The *metadata server logs* contain entries for routine activity and error conditions that are generated by the metadata servers. A separate log is maintained for each metadata server in the cluster. If you access these logs through the master metadata server (either from the SAN File System console or the administrative command-line interface using the **catlog -log cluster** command), you see a consolidated view of the logs for each metadata server in the cluster, ordered by date and time, providing you with cluster-wide view of activities and events. This consolidated view is called the *cluster log*. If you access these logs through a subordinate metadata server (from the administrative command-line interface), you only see the logs for that particular metadata server.

Each metadata server log has a maximum size of 250 MB. Once a log file reaches its maximum size, it is renamed with a .old extension (for example, the log.std file is renamed log.std.old). An existing .old file is overwritten when a subsequent .old file of the same name is created. The log.std file is then cleared and used for new messages of the indicated type. In this way, 500 MB of each type of log data is maintained.

Security log

The *security log* maintains a history of administrator login activity generated by the administrative servers. A separate log is maintained for each administrative server. If you access these logs through the master metadata server (either from the SAN File System console or the administrative command-line interface, you see a

consolidated view of the logs for each administrative servers in the cluster, ordered by date and time. If you access these logs through a subordinate metadata server (from the administrative command-line interface), you see only the logs for the administrative server that is running on the same engine as that particular metadata server.

Trace log

The trace log receives trace messages from the metadata server. Because a minimal amount of tracing is always enabled for first-failure data capture, this log always exists. However, the number of messages and the level of detail that the messages convey is dependent on the current trace settings for the particular metadata server. The default level of tracing that is active at all times is 0, which sends only the most important messages. These messages are useful for providing initial first-failure data capture (FFDC) information. Trace messages are of interest primarily to IBM support personnel. Change the trace settings only at their direction. Higher levels of tracing can generate significant processing activity and should be used only when necessary.

Master console

The *master console* is a serviceability focal point for SAN File System and other IBM TotalStorage products. For SAN File System, the master console provides the key infrastructure for the remote access (through a virtual private network (VPN)) and service alert features.

The master console is a software feature that includes the following software:

- Microsoft Windows 2000 Advanced Server edition
- IBM Director Server
- IBM Tivoli Bonus Pack for SAN Management
- Adobe Acrobat
- The PuTTY openssh package
- Drivers for Qlogic QLA 2342 Fibre Channel Adapter
- VPN Connection Manager
- FAStT Storage Manager

From the master console, you can access the following components:

- SAN File System console, through a Web browser.
- Administrative command-line interface, through a Secure Shell (SSH) session.
- Any of the storage engines in the cluster through an SSH session.
- The RSA II card for any of the storage engines running the SAN File System software through a Web browser. In addition, you can use the RSA II Web interface to establish a remote console to the engine, allowing you to view the engine desktop from the master console.
- Any of the SAN File System clients through an SSH session, a telnet session, or a remote display emulation package (such as VNC), depending on the configuration of the client.

Typically, you use the master console to access the SAN File System console or the administrative command-line interface as well as the storage engines. However, to perform service operations, you can attach a keyboard, monitor, and mouse to an engine if necessary.

Using the remote access feature, you can initiate a VPN connection to allow a support engineer to remotely access the master console. You can monitor that access and disconnect the session at any time.

Metadata server

A *metadata server* is a software server that performs metadata, administrative, and storage-management services and provides clients with shared, coherent access to shared storage (or global namespace). The metadata servers are clustered for scalability and availability, and are often referred to as a cluster. In the cluster, there is one master metadata server and one or more subordinate metadata servers, each running on a separate storage engine. Additional metadata servers can be added, as required, when the workload grows.

All of the metadata servers, including the master metadata server, share the workload of the global namespace. Each is responsible for providing metadata and locks to clients for specific filesets assigned to them. They know which filesets belong to which metadata server, and when contacted by a client, can direct the client to the appropriate metadata server. They also manage distributed locks to ensure the integrity of all of the data within the global namespace.

In addition to providing metadata to clients and managing locks, metadata servers perform a wide variety of other tasks. They process requests to create and manage filesets, storage pools, volumes, and policies; enforce the policies to place files in appropriate storage pools; and send alerts when any threshold established for the filesets and storage pools are exceeded.

Administrative services

The metadata servers process requests from administrators (issued from the SAN File System console or administrative command-line interface) to perform the following types of tasks:

- Create and manage filesets, which are subsets of the entire global namespace and serve as the units of workload assigned to specific metadata servers.
- Create and manage volumes, which are LUNs labeled for SAN File System's use in storage pools.

Data-volume operations are initiated from and coordinated by the metadata servers but are actually performed by one or more clients. These operations are serial and I/O bound, and most have no effect on processing; however, some operations (such as volume drain) are more processor-intensive and might effect client performance. The metadata servers perform metadata-volume operations.

- Create and maintain storage pools. For example, an administrator can create a storage pool that consists of Redundant Array of Independent Disks (RAID) or striped storage devices to meet reliability requirements, or create a storage pool that consists of random-access or low-latency storage devices to meet high performance requirements.
- Create FlashCopy images of filesets in the global namespace that can be used to make file-based backups easier to perform.
- Define policy sets that contain rules that determine in which storage pools specific files are stored.

Metadata services

There are two types of metadata:

- *File metadata* is information that clients need to access files directly from storage devices in the SAN. File metadata includes permissions, owner and group, access time, creation time, and other file characteristics, as well as the location of the file on the storage device.
- *System metadata* is metadata used by the system itself. It includes information about file sets, storage pools, volumes, and policies. The metadata servers perform the reads and writes required to create, distribute, and manage this information. The system metadata is stored and managed in the system storage pool, which is only accessible by the metadata servers in the cluster.

Distributing locks to clients involves the following services:

- Issuing leases that determine the length of time that a metadata server guarantees the locks that it grants to clients.
- Granting locks to clients that allow them shared or exclusive access to files or parts of files. These locks are semi-preemptible, which means that if a client does not contact the metadata server within the lease period, the metadata server can “steal” the client’s locks and grant them to other clients if requested; otherwise, the client can reassert its locks (get its locks back) when it can contact the metadata server again.
- Providing a grace period during which a client can reassert its locks before other clients can obtain new locks if the metadata server itself goes offline and then comes back online.

Metadata servers and filesets

The metadata servers manage such things as file locations, file permissions, and locking. Only the metadata information travels over the IP network, minimizing the data transfer on the IP network. The SAN File System clients still access the regular data in the SAN and thus can benefit from the high performance a SAN can provide. For ideal performance, it is beneficial to evenly balance the workload across the metadata servers. Each fileset is assigned to an metadata server, and thus the filesets can be balanced across the metadata servers in the cluster.

SAN File System provides administrative commands that can be used to monitor transaction rates on each metadata server. Transaction rate parity across all metadata servers provides better SAN File System performance. Fileset assignment can be changed from one metadata server to another to balance transaction rates within the SAN File System environment. You should plan the filesets to be used based on expected I/O transaction rates, because this (rather than file size or storage space consumption) drives workload on the metadata server.

Storage-management services

The metadata servers perform these storage-management services:

- Manage allocation of blocks of space for files on LUNs
- Maintain pointers to the data blocks of the files
- Evaluate the rules in the active policy and manage the placement of files in specific storage pools based on those rules
- Issue alerts when file sets and storage pools reach or exceed their specified thresholds, or return out-of-space messages if they run out of space

SAN File System is designed to optimize communication between the metadata servers and clients. If a client deletes a file, there might be a delay before the free space is visible to the clients. The metadata server periodically checks for and reclaims freed space. Until the space is reclaimed, clients cannot see the freed space.

Remote access

This topic provides an overview of the Remote Access feature and explains the activities required to plan for it.

The master console provides *remote access* to the storage engines. Remote access allows IBM Support Center to diagnose problems with your system. Remote access support can help to greatly reduce service costs and shorten repair times, which in turn reduces the impact of any failures on your business.

Remote access gives IBM support personnel full access to the SAN File System or SAN Volume Controller through the master console, including querying and controlling the metadata servers and clients, and accessing metadata, log, dump, and configuration data. Remote access does not allow access without authentication. You must initiate a secure Virtual Private Network (VPN) connection, using VNC from the master console, to allow IBM support personnel to remotely access the master console. From the master console, the support personnel can establish a connection to the SAN File System metadata servers or SAN Volume Controller nodes. However, you can monitor that access and disconnect the session at any time.

In response to an error condition, you initiate a secure connection to the IBM VPN server using the VPN connection software on the master console called IBM Connection Manager. You must send the customer connection ID for the newly created connection to the IBM support personnel. The IBM support personnel initiates a secure connection to their VPN server, and then establishes a secure connection to the master console over the VPN tunnel using the customer connection ID and an account on the master console. Finally, IBM support personnel can access the SAN File System metadata server or SAN Volume Controller node through SSH.

The following figure shows the remote access architecture:

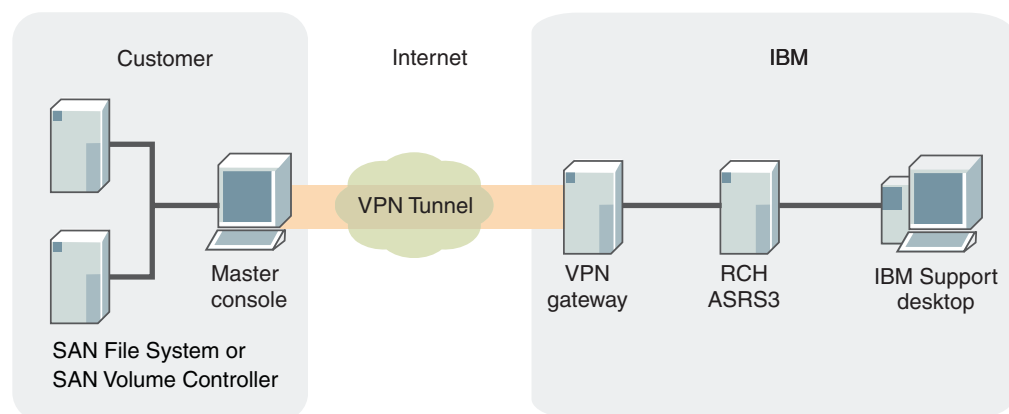


Figure 4. Remote access architecture

Service alert

This topic provides an overview of the service alert.

Service alert is a feature of the master console that enables SAN File System or the SAN Volume Controller to proactively notify the IBM Support Center of significant

errors or failure conditions. This enables IBM to respond quickly to problems that occur, sometimes before the problem has been noticed by your system administrator.

In response to a sever error condition, SAN File System or the SAN Volume Controller issues an Simple Network Management Protocol (SNMP) trap and sends that trap to IBM Director Server running on the master console. The IBM Director Server catches the trap and converts it into a Simple Mail Transfer Protocol (SMTP) e-mail message. The e-mail message is then sent to your SMTP mail server and then forwarded to the IBM support system, where it is converted into a problem record.

This figure shows the service alert architecture:

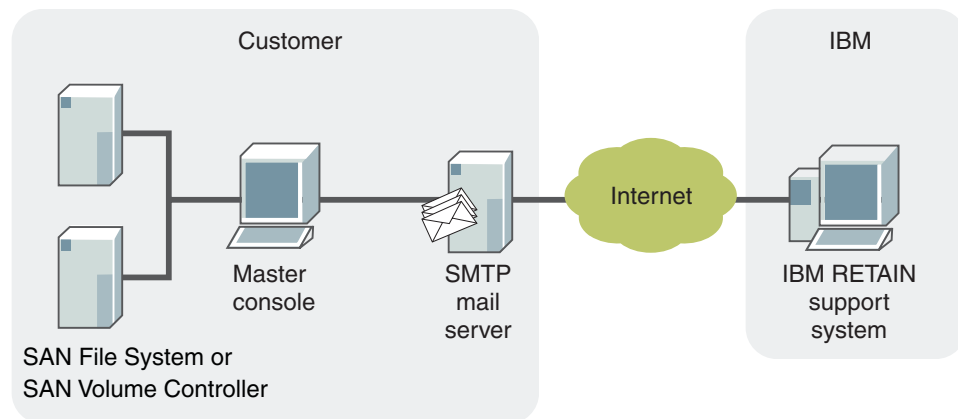


Figure 5. Service alert architecture

SNMP

The *Simple Network Management Protocol (SNMP)* is typically used to monitor network health, and performance and hardware, as well as to find and solve network problems. SNMP consists of two main components:

- SNMP agents, which are software components that reside on managed devices and collect management information (using Management Information Bases or MIBs). SNMP agents issue traps when SNMP events occur. These traps are sent through User Datagram Protocol (UDP) to an SNMP Manager.
- An SNMP manager, which is a network management application (for example, IBM Tivoli NetView®) that monitors and controls devices on which SNMP agents are running and can receive SNMP traps.

In SAN File System, each metadata server generates SNMP traps in response to certain events. SNMP traps are not issued from the operating system, hardware, or the administrative agent.

Tip: The RSA II cards can be set up to generate hardware traps as well.

You can configure which severity levels of events (informational, warning, error, or severe) should generate SNMP traps and you can define which SNMP managers in the SAN environment are to receive the traps. When an event occurs with a severity level that causes an SNMP trap, SAN File System sends the trap, and logs the event in the cluster log.

Note: SAN File System supports asynchronous monitoring through traps but does not support SNMP GETs or PUTs for active management. The SNMP Manager cannot manage SAN File System.

Not all events in SAN File System generate traps. Examples of events that might generate SNMP trap messages include:

- When a metadata server executes a change in state
- When a metadata server detects that another metadata server is not active
- When the size of a file set reaches a specified percentage of its capacity

Storage management

SAN File System provides automatic file placement through the use of policies and storage pools. You can create storage pools that are available to all clients, and define *rules* in policies that cause newly created files to be placed in the appropriate storage pools automatically. For more information about policies, rules, and storage pools, see the related topics below.

File placement

SAN File System provides automatic file placement at the time of creation through the use of policies and storage pools. You can create quality-of-service storage pools that are available to all users and define rules and policies that cause newly created files to be placed in the appropriate storage pool automatically.

A *policy* is a list of rules that determines where the data for specific files is stored. A *rule* is an SQL-like statement that tells a metadata server to place the data for a file in a specific storage pool if the file attribute that the rule specifies meets the specified criteria. A rule can apply to any file being created or to only files being created within a specific fileset depending on how it is defined.

The rules in a policy are processed in order until the condition in one of the rules is met. The data for the file is then stored in the specified storage pool. If none of the conditions specified in the rules of the policy is met, the data for the file is stored in the default storage pool.

Note:

- Rules in a policy are evaluated only when a file is being created. If you switch from one policy to another, the rules in the new policy apply only to newly created files. Activating a new policy does not change the storage pool assignments for existing files. Moving a file does not cause a policy to be applied. You can create multiple policies, but only one policy can be active at a time.
- After a file has been created, you can check its storage pool assignment using the **statfile** command from the administrative command-line interface.
- If you base your policies on user IDs, be aware of how the UNIX **tar** command restores files from backup. During the restore, a file is first created from backup with the user ID of the performer of the backup and is then changed to the user ID of the original creator of the file. With SAN File System, the policy is applied to the file at creation, so the policy applies to the user ID of the performer of the backup rather than to the user ID of the original file creator.

Policies and rules

This topic describes how SAN File System automates the placement of newly created files into storage pools using policies and rules.

Policies and the rules that they contain are used to automatically assign files to specific storage pools.

Policies

A *policy* is a set of rules that determine where specific files are placed based on the file's attributes. You can define any number of policies, but only one policy can be active at a time. If you switch from one policy to another or make changes to a policy, that action has no effect on existing files in the global namespace. The new or changed policy is effective only on newly created files in SAN File System. Moving a file does not cause the policy to be applied.

A policy can contain any number of rules. There is no limit to the size of a policy.

SAN File System performs error checking for policies in the following phases:

- When you create a new policy, the master metadata server checks the basic syntax of all the rules in the policy.
- When you activate the policy, the master metadata server checks all references to filesets and storage pools. If a rule in the policy refers to a fileset or storage pool that does not exist, the policy is not activated and an error is returned.
- When a new file is created by a client, the rules in the active policy are evaluated in order. If an error is detected, the metadata server responsible for creating the file logs an error, skips all subsequent rules, and assigns the file to the default storage pool. If a default pool does not exist, the file is not created and the metadata server returns an error to the client application.

If your environment is set up in a non-uniform zone configuration (in which clients cannot access all volumes), you need to ensure that the rules in the active policy place files into volumes that are accessible to the clients that use them.

Tip: When SAN File System is first installed, a default policy is created and remains active until you create and activate a new one. The default policy assigns all files to the default storage pool. Although the default storage pool is created when SAN File System is first started, you must assign volumes to it before it can be used. If a user or application on a SAN File System client attempts to create new files that would be assigned to the default storage pool, and there are no volumes assigned to it, the user or application receives No Space errors.

Rules

A *rule* is an SQL-like statement that tells the metadata server to place the data for a file in a specific storage pool if the file meets specific criteria. A rule can apply to any file being created or only to files being created within a specific fileset or group of filesets.

Rules can specify any of these conditions, which when matched, causes that rule to be applied:

- Fileset
- Filename or extension
- Date and time when the file is created

- User ID and Group ID on UNIX clients

SAN File System evaluates rules in order, from top to bottom, as they appear in the active policy. The first rule that matches determines the file's placement. In other words, when a client creates a file, SAN File System scans the list of rules in the active policy to determine which rule applies to the file. When a rule applies to the file, SAN File System stops processing the rules and assigns the file to the appropriate storage pool. If no rule applies, the file is assigned to the default storage pool.

Tip:

- After a file has been created, you can check its storage-pool assignment from the administrative command-line interface using the **statfile** command.
- You can use the **statpolicy** command from the administrative command-line interface to view the statistics about the policy rules.
- If you base your policies on user IDs, be aware of the manner in which the UNIX **tar** command restores files from backup. During the restore, a file is first created from backup with the user ID of the performer of the backup and is then changed to the user ID of the original creator of the file. With SAN File System, the policy is applied to the file at creation, so the policy applies to the user ID of the performer of the backup rather than to the user ID of the original file creator.
- During a restore or migration, a rule that uses the creation date as the placement criteria will assign a file based on the time of the restore or migration, not the original creation time, and a rule that uses a user ID or group ID as the placement criteria will assign a file based on the user ID or group ID of the restore or migration. Therefore, do not use creation time, user ID or group ID to place file.

For detailed information about creating policies and rules, see the related topics below.

Storage subsystems

SAN File System conforms to small computer system interface (SCSI) standards and is designed to work with any SCSI-compliant storage devices, including Just a Bunch Of Disks (JBOD), redundant array of independent disks (RAID) with mirroring, and hierarchically-managed storage devices. You can attach tape devices to SAN File System for backups and long-term storage, although tape devices cannot be part of a storage pool.

All storage subsystems attached to SAN File System can be accessed by all clients (unless you use zoning to allow only specific clients to access specific devices). This enables data sharing among heterogeneous clients.

SAN File System supports heterogeneous, simultaneously-connected storage and host-bus adapter (HBA) sharing, subject to client platform, driver, and storage-vendor limitations.

Currently, SAN File System supports only these storage subsystems for use in the system storage pool:

- The IBM TotalStorage Enterprise Storage Server (ESS), models 2105-750
- The IBM TotalStorage SAN Volume Controller, model 2145 with storage subsystems that are supported by SAN Volume Controller

- IBM FAStT 600T, 700, and 900 running firmware version 8.4 on the storage device and software version 8.41 on the client platforms

Restriction:

- SAN File System is designed to work with any storage device that is used in a device driver-operating system stack and that conform to the SCSI standard, including the unique-device-identification capability. non-IBM storage subsystems for user storage pools (data volumes); however, you must use IBM storage subsystems for the system storage pool (metadata volumes).
- Storage subsystems other than ESS or SAN Volume controller require additional, manual configuration to be detected and used by SAN File System. You must create a character-special device file for each device to be used if the operating system or device driver has not already done so. You must also create a soft link to this device file in a SAN File System-designated directory in the device hierarchy.

Storage pools

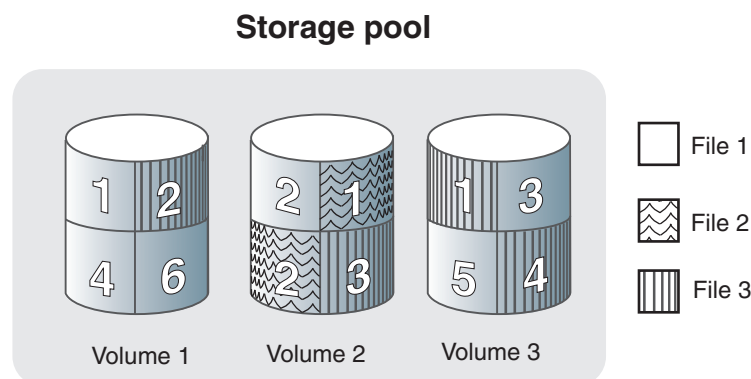
A *storage pool* is a named set of SAN File System volumes that can be used to store either metadata or file data. A storage pool consists of one or more volumes that provide a quality of service that you want for a specific use, such as to store all files for a particular application or a specific business division. You must assign one or more volumes to a storage pool before it can be used.

SAN File System has two types of storage pools: system storage pool and user storage pool.

Storage pools and volumes

Typically, you assigns volumes to storage pools based on their common characteristics, such as device capabilities (availability or performance level) and usage (business division, project, application, location, or customer).

Each storage pool manages its own volumes. File space is allocated to the volumes in a given storage pool in a round-robin algorithm (as shown in the following figure) in logical partitions, or in blocks. Logical partitions are allocated to the system storage pool in 16-MB blocks. For user storage pools, including the default storage pool, you can allocate logical partitions in 16, 64, or 256-MB blocks. All logical partitions in the same storage pool must be the same size.



Tip: You can set a threshold to generate an alert when a storage pool reaches or exceeds a certain percentage of its maximum capacity. By default, an alert is

generated when a storage pool becomes 80% full. An alert is logged every five minutes until one or more volumes are assigned to the storage pool. You can set configuration parameters to cause an SNMP trap message to be generated as well. An SNMP trap notifies you of this condition asynchronously.

System storage pool

The *system storage pool* contains the system metadata (system and file attributes, configuration information, and metadata server state) that is accessible to all metadata servers in the cluster. There is only *one* system storage pool that is created automatically when SAN File System is installed. The system storage pool contains the most critical data for SAN File System. The first volume that is assigned to the system storage pool, called the *master volume*, contains the most critical pages of metadata that SAN File System manages.

Important: Use highly-reliable and available logical unit numbers (LUNs) for the system storage pool (for example, mirroring or redundant array of independent disks (RAID), plus hot spares in the backend storage system) so that the cluster always has a robust copy of the system metadata.

Because the amount of metadata grows as the global namespace grows, you must monitor the system storage pool to ensure that there is always enough volumes assigned to it to accommodate the growth. The system storage pool typically requires approximately 2% to 5% of the total storage capacity that SAN File System manages, but this amount varies depending on your environment. Use the alert features on the system storage pool to ensure that you do not run out of space.

Tip: The minimum size of a system volume is 2 GB; therefore, the minimum size of the system storage pool is also 2 GB.

For security and reliability, the volumes that are assigned to the system storage pool should be accessible only to the cluster using a private SAN or a shared SAN with a combination of zoning, LUN masking, or special configuration. For reliability, the volumes should be virtualized RAID arrays (also known as *ranks* within IBM Enterprise Storage Server).

User storage pools

A *user storage pool* contains the blocks of data that make up user files. SAN File System stores the data that describes the files, called file metadata, separately from the actual file data. You can create one or more user storage pools, and then create policies that contain rules that cause metadata servers to store data for specific files in the appropriate storage pools.

The *default storage pool* is a special user storage pool. This optional storage pool is used to store the data for a file if the file is not assigned to a specific storage pool by a rule in the active policy. A default storage pool is created when SAN File System is installed. However, if you want to use the default storage pool, you must assign one or more volumes to it. There can be only one default user storage pool in SAN File System. You can designate any user storage pool that has volumes assigned to it to be the default storage pool. You can choose to disable the default storage pool. In this case, newly created files that do not match any rules in the active policy are not saved.

User interfaces

There are two methods for managing SAN File System: an *administrative command-line interface* and a graphical user interface, called the *SAN File System console*. You can access the administrative command-line interface by either directly logging in to an engine or using a Secure Shell (SSH) client to remotely connect to the engine. You can access the SAN File System console using a Web browser.

SAN File System provides you with different levels of user access to perform administrative operations. The users and user roles are defined on your LDAP server. Therefore, you cannot access the SAN File System without a valid user ID that is defined in the LDAP server.

SAN File System provides the following user interfaces:

- A Web-based administrative user interface called the SAN File System console
- An administrative command-line interface
- A client command-line interface

Note: The administrative server does not lock administrative access in order to prevent simultaneous SAN File System console or administrative command-line interface sessions. You must manually coordinate the use of the administrative interfaces.

SAN File System console

The SAN File System console allows you to control and monitor SAN File System from a Web-based graphical user interface. For ease of monitoring, it provides a system overview that illustrates the status of the various SAN File System components. In addition, the SAN File System console provides inline messaging that assists with system configuration, performance tuning and troubleshooting tasks.

The SAN File System console also contains the Help Assistant, which provides panel-level help information as well as links to related topics in the SAN File System Information Center. The Information Center serves as an online, searchable repository for all of the product documentation.

Administrative command-line interface

You can use the administrative command-line interface to administer all aspects of SAN File System, including setting up and managing storage pools, volumes, and filesets. For security reasons, administrative command-line interface runs only on the engines in your cluster.

You can use the administrative command-line interface interactively using the **sfscli** utility. You can also embed administrative commands in scripts.

To access **sfscli**, you must log in to an engine that hosts any metadata server. The following figure illustrates how you access **sfscli**.

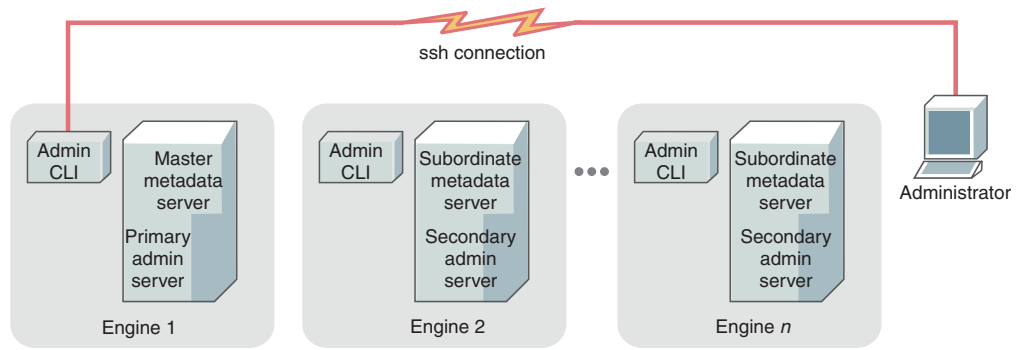


Figure 6. Accessing `sfsccli`

Client commands

SAN File System provides a set of commands that are used to set up SAN File System clients and to perform planning, migration, and verification tasks for data. These commands are issued from the client machines.

Volumes

A *logical unit number* (LUN) is the logical unit of storage that a SAN or other disk subsystem can assign to metadata servers and clients. A *volume* is a LUN that is labeled by SAN File System for its use. Volumes are grouped together virtually to form storage pools, in which file data and metadata is stored.

An LUN becomes a SAN File System volume when you add it to a storage pool. It is automatically assigned a system-generated label that identifies it as a SAN File System volume. You must also give the volume a name that is unique among all the volumes used by a SAN File System cluster.

During startup, the metadata server scans all LUNs that it can access in the SAN, searching for the label that tells it that the LUN is a valid SAN File System volume. Clients perform this same search whenever they are started.

System-data LUN operations are performed by the metadata servers. All other data LUN operations are initiated from and coordinated by the metadata servers in the cluster but are actually performed by one or more clients; therefore, the metadata servers no longer need to see the data LUNs, and the clients only need to see the data LUNs that they need to access. This allows SAN File System to support a wide variety of SAN configurations, storage devices, and drivers, and also supports scaling to large numbers of storage devices and clients. This also allows SAN File System to support grouping clients and LUNs into SAN zones to provide enhanced security.

A volume must be empty to be removed from a storage pool. When you remove a volume, SAN File System moves (drains) the contents of that volume across other available volumes in the same storage pool. If the storage pool does not have sufficient space available in other volumes to move all of the data contained in the specified volume, the removal fails and the metadata server suspends the volume (the metadata server cannot allocate new data on that volume).

Tip:

- Keep the storage subsystem device driver's virtual path (vpath) configuration file current. If many LUNs are added and deleted from the metadata server, it is possible for the configuration file to contain references to LUNs that do not exist.
- When the number of entries in the storage subsystem device driver's vpath configuration file reaches 255, any new LUN configured on the metadata server will not be visible.

Volume activation and suspension

When you add a volume to a storage pool, by default, the volume is activated. This means that a metadata server can allocate data to the newly added volume. You can add a volume to a storage pool in a suspended state; however, no data can be allocated to the volume until you activate it.

A volume can be in a suspended state if you add it to a storage pool without activating it. You can also change the state of a volume from activated to suspended. When a volume is in a suspended state, a metadata server cannot allocate any data to it.

Volumes and storage pools

When you install SAN File System, there is a system storage pool, which is used by metadata servers to store system and file metadata, and a default storage pool, which can be used to store file data. You can create additional user storage pools for file data; however, no data can be stored in a storage pool until you assign one or more volumes to it. You can also remove the default storage pool if you choose.

The volumes added to the system storage pool are called *system volumes*.

As the amount of metadata that is generated for the server cluster and client files grows, you must ensure that the system storage pool always has enough volumes assigned to it so that it does not run out of space.

You must also ensure that any user storage pools, including the default storage pool, has a sufficient number of volumes. Each storage pool must have at least one volume assigned to it before any files can be stored in it.

To assist you in monitoring storage pool capacity, SAN File System provides a threshold option that you can specify when adding a volume to a storage pool or changing settings for a storage pool. A threshold is a specified percentage of the estimated maximum capacity of the storage pool. When a storage pool reaches or exceeds the percentage specified as its threshold, SAN File System generates an alert. This alert can also generate an SNMP trap message to notify you of the condition asynchronously, if you set the appropriate parameters for SNMP traps.

Volume considerations

- Removing a volume could impact performance if that volume contains a large amount of data.
- In a non-uniform zone configuration, you create multiple zones in which clients have access to only the volumes that they will actually need. If your environment uses a non-uniform zone configuration, you must ensure that the clients can access all of the volumes in any user storage pool that can be used by a filesets in use by that client. If a client tries to read or write data on a volume that it cannot access, SAN File System will return an I/O error. File system operations that involve only metadata, such as changing the current directory or listing files, will not receive an I/O error because those functions do not require access to the user storage pool.

Volume removal

You can remove a volume from a storage pool. During this process, any files that are stored in the volume are automatically redistributed among the remaining volumes in the same storage pool. When you remove a volume, data is moved and committed one logical partition at a time. If a failure occurs while moving the contents of a volume, you can reissue the command, and the move process continues where it stopped earlier.

If I/O errors occur for some files or if the redistribution of the files to other volumes fails, you can specify a force option that causes the volume to be removed anyway. If there is a bad file on the volume, specifying this option causes the system to delete the entire file, even if parts of the file reside on other volumes.

If the bad file is part of a FlashCopy image, SAN File System removes it from the FlashCopy image. This removal might render the FlashCopy image non-reversible. You can still backup and restore the remaining files on the FlashCopy image.

Remember: There is no automatic recovery process when you specify the force option. All files are removed immediately without being copied. Before removing a volume with the force option, use the **reportvolfiles** command to display a list of files on the volume. The files for which failures occur are also listed in the cluster message log, and you can restore those files manually.

When you remove a volume from a storage pool, its label is removed, and the volume becomes a logical unit number (LUN) again.

Limitations to volumes in the system storage pool

The volumes in the system storage pool have these limitations:

- All volumes in the system storage pool must be of the same type of backend storage device and must be one of the supported IBM storage subsystems. You can use IBM TotalStorage SAN Volume Controller to provide mixed storage as long as only the SAN Volume Controller virtual devices are visible to the cluster.
- All volumes in the system storage pool must be visible to all metadata servers in the cluster.
- Each volume in the system storage pool must be at least 2 GB in size.
- The system storage pool is limited to 126 dual-path volumes.

Chapter 2. Accessing the administrative interfaces

This section discusses the tasks for accessing the SAN File System console and administrative command-line interface.

Accessing the administrative command-line interface

This topic describes how to bring up the administrative command-line interface.

SAN File System provides a wizard to step you through the process of creating a FlashCopy image.

You must have Administrator, Operator, or Backup privileges to perform this task.

1. Log in directly to the engine, or from another workstation through SSH, using the local operating system authentication mechanism.
2. Log in to the administrative server on the engine using the same administrative user ID and password that you would use to log into the SAN File System console. You can specify the password in one of two ways:
 - Set the password in the `sclif.properties` file, located in your home directory on the engine (for example, `joe/sclif.properties`), to your valid LDAP password using the **tankpasswd** utility.
 - Set the `SFS_CLI_PASSWDFILE` environment variable to the location of the password file.
3. Enter the **sfscli** command to start the administrative command-line interface, or **sfscli** session, to run commands in interactive mode.

Accessing the SAN File System console

This topic describes how to bring up the SAN File System console from the master console.

1. Open a supported Web browser window and enter the URL:
`https://master_metadata_server:port/sfs`, where *master_metadata_server* is the name or IP address of the master metadata server, and *port* is the port number of the master metadata server. The default port number is 7979.
2. When the SAN File System signon screen displays, enter your administrator ID and password.

Chapter 3. Monitoring

You can monitor the SAN File System by displaying statistics and status.

Creating a report

This topic describes how to create a report of component statistics.

1. Click **Monitor System**→**Statistics** from the My Work frame.
2. Click the **Create Report** tab.
3. Select the components for which you want to include statistics in the report.
4. Click **Create Report**.

Note: You can print this report using the print function in your Web browser.

5. Click **Close** to close the report.

Monitoring metadata server HBA statistics

This topic describes how to monitor metadata server host-bus adapter (HBA) performance data.

You must have Administrator privileges to perform this task.

1. Log in to the metadata server.
2. Enter the **datapath query adaptstat** command at the Linux prompt to monitor metadata server Fibre Channel (FC) adapter performance.
`<servername>:~ # datapath query adaptstat`

Viewing metadata server system statistics

This topic describes how to use standard Linux utilities, which are provided on the metadata server operating system, to view system statistics.

You must have Administrator privileges to perform this task.

1. Use the **top** command to display, in real time, the most active CPU processes on the system. These statistics include its process ID (PID), memory usage, swap file usage, and uptime (how long it has been running).
 - a. Log in to the metadata server.
 - b. Enter **top** at the Linux prompt.
`<servername>:~ # top`
2. Use the **vmstat** command to display, in real time, the performance information summarized at the system level. These statistics include the number of processes, paging information, and the percentage of CPU activity broken down by user, system, and idle time.
 - a. Log in to the metadata server.
 - b. Enter **vmstat** at the Linux prompt.
`<servername>:~ # vmstat`

Viewing SAN File System component statistics

This topic describes how to view statistics for specific SAN File System components.

1. Click **Monitor System→Statistics** from the My Work frame.
2. Click the tab for the component for which you want to view statistics.
3. Click **OK** to view the statistics.
4. Click **Close** to close the Statistics panel.

Viewing system status

This topic describes how to view status of the major SAN File System components.

To view status for the major SAN File System components (including cluster state, metadata server activity and availability, and recent messages), click **Monitor System→System Overview** from the My Work frame.

Note: The SAN File System console provides limited filtering for recent messages. You can view either all messages or only the alerts that have been generated. To change the type of messages that is displayed, select the filter from the **Message Type** drop-down list, and then click **Submit**.

You can also change the interval at which the information on this page is refreshed. To change the refresh rate or to disable refreshing, select the appropriate interval from the **Refresh Interval** drop-down list, and then click **Submit**.

Chapter 4. Managing

This section discusses tasks that will help you manage SAN File System on a daily basis.

Managing administration properties

As part of administrative security, an administrator must define users, their authenticating passwords, and SAN File System administration roles on an LDAP server. An administrator also maintains a Remote Service Adapter (RSA) user ID and password that is used to access information from the engines. These RSA and LDAP settings are known as administration properties.

Changing LDAP configuration settings

This topic describes how to change LDAP configuration settings.

You must have Operator or Administrator privileges to perform this task.

1. Click **Administrator Access**→**Administration Properties** from the My Work frame.
2. Click the **LDAP Access** tab.
3. Edit the appropriate fields.
4. Click **OK** to save the new settings.

Changing RSA userid and password

This topic describes how to change the RSA userid and password, which are used to access information from engines. The values must match those in the RSA system.

You must have Operator or Administrator privileges to perform this task.

1. Click **Administrator Access**→**Administration Properties** from the My Work frame.
2. Click the **System Access** tab.
3. Edit the appropriate fields.
4. Click **OK** to save the new settings.

Viewing administration properties details

This topic describes how to view administration properties, which are LDAP settings and the RSA user ID and password.

You must have Operator or Administrator privileges to perform this task.

1. Click **Administrator Access**→**Administration Properties** from the My Work frame.
2. Click the **Details** tab.

Managing alerts and logs

Through the setting of alerts, and SNMP traps, as well as the viewing of logs, you can monitor SAN File System components such as storage pool size, metadata server performance, and so forth. The following topics will assist you with configuring SAN File System alerts and SNMP managers and in viewing and clearing logs.

Adding SNMP managers

This topic describes how to add SNMP managers. An SNMP manager is an SNMP server, which forwards traps to the appropriate location.

You must have Administrator privileges to perform this task.

The SAN File System console allows you to specify up to two SNMP managers.

Note: If Service Alert is configured, then one of the managers will be set to the Service Alert SNMP client.

1. Click **Monitor System**→**SNMP Properties** from the My Work frame.
2. Click the **SNMP Managers** tab, if it is not already selected.
3. Type a **Destination IP** for one or both managers.
4. Type a **Destination Port** for one or both managers.
5. Select the appropriate **SNMP version** for one or both managers.
6. Type an **SNMP community** for one or both managers.
7. Click **OK** to confirm the addition of the manager or managers specified.

Clearing logs

This topic describes how to clear the audit log or cluster log.

You must have Administrator privileges to perform this task.

You can clear the entire audit log or cluster log.

Note:

1. Clearing the cluster log also clears the event log because the event log is a subset of the cluster log.
 2. You cannot clear the administrative log or security log through the SAN File System console or the administrative command-line interface.
1. Click **Monitor System** and then click **Audit Log** or **Cluster Log** from the My Work frame.
 2. Click the **Clear Log** button above the log content.
 3. Click **Clear Log** to clear the log.

Deleting SNMP managers

This topic describes how to delete SNMP managers. An SNMP manager is an SNMP server, which forwards traps to the appropriate location.

You must have Administrator privileges to perform this task.

The SAN File System console allows you to specify up to two SNMP managers. The SNMP manager is really an SNMP server, which then forwards traps to the appropriate location (such as an SNMP client if one is present in your environment).

1. Click **Monitor System**→**SNMP Properties** from the My Work frame.
2. Click the **SNMP Managers** tab, if not already selected.
3. Delete the text in the **Destination IP** field or fields.
4. You can also delete information in the other fields for the associated manager, but it is not required.
5. Click **OK** to confirm the deletion of the manager.

Modifying SNMP managers

This topic describes how to modify SNMP managers. An SNMP manager is an SNMP server, which forwards traps to the appropriate location.

You must have Administrator privileges to perform this task.

The SAN File System console allows you to specify up to two SNMP managers. The SNMP manager is really an SNMP server, which then forwards traps to the appropriate location (such as an SNMP client if one is present in your environment).

1. Click **Monitor System**→**SNMP Properties** from the My Work frame.
2. Click the **SNMP Managers** tab.
3. Modify the information in the fields for one or more managers as required.
4. Click **OK** to confirm the changes.

Modifying SNMP traps

This topic describes how to modify the event severity levels that get specified as SNMP traps.

You must have Administrator privileges to perform this task.

1. Click **Monitor System**→**SNMP Properties** from the My Work frame.
2. Click the **SNMP Events** tab.
3. Change the selected event severity levels for which you would like SNMP traps to be sent.
4. Click **OK** to confirm the changes.

Setting up SNMP traps

This topic describes how to select the event severity levels that get specified as SNMP traps.

You must have Administrator privileges to perform this task.

1. Click **Monitor System**→**SNMP Properties** from the My Work frame.
2. Click the **SNMP Events** tab.
3. Select the event severity levels for which you would like SNMP traps to be sent.
4. Click **OK** to confirm the selected levels.

Viewing logs

This topic describes how to view status for the SAN File System components.

SAN File System console allows you view these logs to help you monitor the system and troubleshoot problems:

- Administrative log
- Audit log
- Cluster log
- Event log
- Security log

The cluster log is a consolidated view of the server log on each engine in the cluster. The event log is a subset of the messages stored in the cluster log; it contains only those messages that have an message type of event.

1. Click **Monitor System** from the My Work frame.
2. Click **Cluster log**.
3. Select **Events only** from the **Type** drop-down list.
4. Click **Refresh**.

Viewing SNMP managers

This topic describes how to view SNMP managers. An SNMP manager is an SNMP server, which forwards traps to the appropriate location.

The SAN File System console allows you to specify up to two SNMP managers. The SNMP manager is really an SNMP server, which then forwards traps to the appropriate location (such as an SNMP client if one is present in your environment).

1. Click **Monitor System**→**SNMP Properties** from the My Work frame.
2. Click the **SNMP Managers** tab.
3. The panel displays the information about currently configured SNMP managers.
4. Click **Cancel** to close the window, or the close button to close the window.

Managing backups

SAN File System supports the use of backup tools that are already present in your environment. For example, if your enterprise currently uses a storage management product such as Tivoli® Storage Manager (TSM), you can use the functions and features of that product to back up and restore files that reside in the SAN File System global namespace.

For backing up in a normal, available environment, you can use the FlashCopy image feature of SAN File System.

To prepare for disaster recovery in situations where SAN File System becomes unavailable, you can perform LUN-based backups using the instant copy features that exist in the storage subsystems that SAN File System supports. If your SAN storage subsystems do not offer copy services, you must back up for disaster recovery using the API method.

Backing up using the LUN method

This topic describes how to perform SAN File System backup operations using the LUN method. LUN backup requires that all transactions are stopped during the process.

The LUN method of backup is only available to SANs comprised of storage subsystems with built-in copy services. SANs without such service must use the file-based (API) method of backup.

Because the LUN method deals with data at the byte level, it is an all-or-nothing approach for backing up and restoring your entire SAN File System. In particular, it provides no ability to restore individual files (because it has no concept of files); you have to save and restore all the data — metadata and file data — or none of it. Restoring a previously saved FlashCopy image is the best method for recovering some subset of SAN File System data. Therefore, the LUN method is best employed as part of a disaster recovery situation.

1. Stop or pause all SAN File System client applications. Because this task is application-specific, refer to the application documentation for details on performing this step.
2. The metadata server and all clients must complete all active transactions and flush their data to disk. From the master metadata server, quiesce the SAN File System metadata servers.

```
/usr/tank/admin/bin/sfsccli quiescecluster -state full
```

This procedure will also lock out any subsequent new I/O from the clients or metadata server.

3. Initiate the storage subsystem copy service using the procedure defined in its accompanying documentation.
4. After the storage subsystem copy is complete, re-enable the SAN File System metadata servers.

```
/usr/tank/admin/bin/sfsccli resumecluster
```

This procedure will also lock out any subsequent new I/O from the clients or metadata server.

5. Restart the client applications using the specific procedures for those applications.

For additional information about restore procedures, including commands, refer to the *IBM TotalStorage SAN File System Maintenance and Problem Determination Guide*, on the publications CD that came with your metadata servers.

Backing up using the file-based (API) method

This topic describes how to perform SAN File System backup operations using two variations of the file-based (API) method.

The file method of backup is used for SANs consisting of storage subsystems that do not offer built-in copy services. SANs that do offer copy services can use the LUN method of backup.

You have two possible options when using the file method of backup. Which method you choose depends on the characteristics of the backup application in your existing environment.

- If your existing backup application allows you to selectively choose subdirectory branches for backup, and allows you to restore files to the directory two levels above their original location, follow this optimized procedure for SAN File System file-based backup:
 1. Stop or pause all SAN File System client applications. Because this task is application-specific, refer to the application documentation for details on performing this step.
 2. Create FlashCopy images of each fileset.

```
/usr/tank/admin/bin/sfsccli mkimage -fileset fileset_name  
-dir directory_name Flashcopy_image_name
```

Tip: Consider using the SAN File System console. It allows you to create multiple FlashCopy images quickly and easily. See Saving a FlashCopy image of a fileset.

3. From the master metadata server engine, save the most recent metadata to accompany the FlashCopy images.

```
/usr/tank/admin/bin/sfsccli mkdrfile most_recent_metadata_file_name
```

A text file is created in /usr/tank/server/DR. Copy the resulting file onto your backup medium (usually tape).

4. On each metadata server backup additional operating system and SAN File System administration configuration files.

```
/usr/tank/admin/bin/setupsfs -backup
```

A tar file called *DRfiles-metadata_server_name-timestamp.tar.gz* is created in /usr/tank/server/DR/ (by default). Copy the resulting tar file onto your backup medium (usually tape).

5. Restart the client applications using the specific procedures for those applications.
6. Use the backup application to back up all *fileset_name/directory_name* subdirectories and their contents, along with all of the *DRfiles-metadata_server_name-timestamp.tar.gz* files to your backup medium (usually tape).

Attention: Backup Windows filesets only from a Windows client; backup AIX filesets only from an AIX client.

- If your existing backup application does not provide the features required for the enhanced method, then follow this procedure for SAN File System file-based backup:

Note: The enhanced method allows you to re-enable client applications more quickly. Creating FlashCopy images is much quicker than backing up the real files to tape.

1. Stop or pause all SAN File System client applications. Because this task is application-specific, refer to the application documentation for details on performing this step.

2. From the master metadata server engine, save the most recent metadata.

```
/usr/tank/admin/bin/sfsccli mkdrfile most_recent_metadata_file_name
```

A text file is created in /usr/tank/server/DR/. Copy that file onto your backup medium (usually tape).

3. Use the backup application to backup all *fileset_name/directory_name* subdirectories and their contents, to your backup medium (usually tape). If possible, exclude all .flashcopy subdirectories and their contents because they will not be of any use during a subsequent restore operation.

4. On each metadata server backup additional operating system and SAN File System administration configuration files.

```
/usr/tank/admin/bin/setupsfs -backup
```

A tar file called *DRfiles-metadata_server_name-timestamp.tar.gz* is created in /usr/tank/server/DR/ (by default). Copy the resulting tar file onto your backup medium (usually tape).

5. Restart the client applications using the specific procedures for those applications.

For additional information about restore procedures, including commands, refer to the *IBM TotalStorage SAN File System Maintenance and Problem Determination Guide*, provided on the publications CD that came with your metadata servers.

Managing clients

A SAN File System client is a computer that accesses and creates data that is stored in the SAN File System global namespace. Some client-management tasks are performed from the SAN File System through the SAN File System console and Administrative command-line interface, and some tasks are performed from the client shell.

Accessing a client through SSH

This task describes how to remotely access a client using SSH from the master console.

Before accessing a client using SSH, the following requirements must be met:

- An operating system account must be set up for use in signing on to the client to be accessed.
 - An SSH software package, such as PuTTY, must be installed and running on the client to be accessed.
 - If the service representative is accessing the master console remotely, you must have previously initiated a VPN connection with the service representative.
1. From the master console, use one of these methods to access the client:
 - a. Open a shell prompt and type `putty.exe -ssh client_IP_address`, where *client_IP_address* is the IP address of the client to be accessed.

Note: If you used SSH to establish a remote session with the master console, type this command from that session to establish an SSH session between the master console and the client.

- b. Click **Start>Programs>PuTTY>PuTTY**.
 - 1) Type the IP address of the client to be accessed.
 - 2) Select SSH as the protocol.
 - 3) Click **Open**.
 2. After the session is established, you can log in using an operating system user ID and password.

After the connection is established, you can perform these activities:

- If you are accessing an AIX client, you can run SAN File System commands that provide the ability to stop and start the client as well as list client status. If you are accessing a Windows client, you can run the `migratedata` command.
- Access operating-system commands to enable or disable tracing, obtain dumps, and stop or start applications.

Accessing a client through telnet

This task describes how to remotely access a client using telnet from the master console.

Before accessing a client using telnet, the following requirements must be met:

- An operating system account for the support representative must be set up on the client to be accessed.

- If the service representative is accessing the master console remotely, the customer must have previously initiated a VPN connection with the service representative.
- 1. From the master console, open a shell prompt and type **telnet *host***, where *host* is the IP address of the client to be accessed.
- 2. After the session is established, you can log in using an operating system user ID and password.

After the connection is established, you can perform these activities:

- If you are accessing an AIX client, you can run SAN File System commands that provide the ability to stop and start the client as well as list client status. If you are accessing a Windows client, you can run the `migratedata` command.
- Access operating-system commands to enable or disable tracing, obtain dumps, and stop or start applications.

Accessing a client through a remote console utility

This task describes how to remotely access a client using a remote console utility, such as VNC, from the master console.

Before accessing a client using a remote console utility, such as Virtual Network Computing (VNC), the following requirements must be met:

- An operating system account must be set up on the client to be accessed.
- A VNC server must be installed and running on the client to be accessed. In addition, a session password must exist.
- A VNC client must be installed and running on the master console.
- If the service representative is accessing the master console remotely:
 - You must have previously initiated a VPN connection with the service representative.
 - The service representative must have established the VPN connection and used a remote display emulation package, such as VNC, to remotely view the desktop of the master console.
- 1. Double-click the VNC viewer icon on the master console.
- 2. Enter the IP address or host name of the client to be accessed and click **OK**.
- 3. Enter the session password and click **OK**.

After the connection is established, you will see the remote desktop of the client. From the VNC connection, you can perform tasks and run commands as if you were physically at the client machine.

Displaying the client-driver version

This topic describes how to display the client-driver version.

1. For AIX, to display the version of a client driver, run the **stfsstatus** command from the client shell.
2. For Linux clients, run the command **cat /proc/fs/stfs/status** from the client shell.
3. For Windows, select Properties on the `stfs.sys` driver and select the Version tab, specifically:
 - a. Open Windows Explorer.
 - b. Locate the SAN File System loadable kernel module, `stfs.sys`.
 - c. Right click on it and select Properties.

- d. Select the Version tab in the Properties page to view the release and source versions.

Downloading client software

This topic describes how to download client software.

You must be signed on to the SAN File System console from the local machine on which you want to download the client software.

This command downloads client software from the `/usr/tank/packages` directory on the engine running the master metadata server.

1. Click **Maintain System**→**Download Client Software** from the My Work frame.
2. Click the version of client software that you want to download, based on the operating system and version.

Enabling client awareness of LUN expansion

This topic describes how to ensure that a client is aware of SAN Volume Controller LUN expansion.

For AIX 5.2 clients, you can perform the following steps to ensure that the client recognizes the new LUN size:

1. Temporarily unmount the client file system.
2. Create the client again.

Granting root privileges to a client

This topic describes how to grant root privileges to a client.

You must Operator or Administrator privileges to perform this task from the SAN File System console. You must have Administrator privileges to perform this task from the administrative command-line interface.

1. Click **Manage Servers and Clients**→**Client Sessions** from the My Work frame.
2. Select one or more client sessions to which you want to grant privileges.
3. Click **Grant Clients Root Privileges** from the drop-down box in the table header.
4. Click **Go**.
5. Click **Grant Root Privileges** to grant privileges to the specified clients.

Implementing file sharing - UNIX fileset on a Windows client

This topic describes how to share UNIX filesets on a Windows client.

You must have Administrator privileges to perform this task.

1. Ensure that you are logged on to a client with Administrator privileges.
2. Take ownership of the UNIX fileset.
3. Run the `ls -l` command to list the permissions.
4. Issue the `chmod` command at the client prompt to set the UNIX permissions to 755.

Now the owner (root) has read, write, and execute permissions, members of the system group have read and execute permissions, and everyone else (including Windows clients) has read and execute permissions.

Implementing file sharing - Windows fileset on a UNIX client

This topic describes how to share Windows filesets on a UNIX-based client.

You must have Administrator privileges to perform this task.

1. Ensure that you are logged on as an administrator or a member of the administrator's group.
2. Open Windows Explorer.
3. Take ownership of the Windows fileset.
4. Select the attached fileset.
5. Right-click and select **Properties**.
6. Select the **Security** tab.
7. Set the permissions on Windows for the "everyone" group.
8. Click **OK**.

Listing client sessions

This topic describes how to display a list of the current client sessions.

To display a list of the current client sessions from the SAN File System console, click **Manage Servers and Clients**→**Client Sessions** from the My Work frame.

Listing clients that can see a LUN

This topic describes how to retrieve a list of clients that can see a specific data LUN.

You must have Administrator privileges to perform this task from the SAN File System console.

1. Click **Manage Storage**→**Data LUNs** from the My Work frame.
2. Select the data LUN from the table.
3. Click **Clients that can see the LUN** from the drop-down box in the table header.
4. Click **Go**.

Listing clients with root privileges

This task describes how to display a list of clients with root privileges.

To display a list of clients with root privileges, click **Manage Servers and Clients**→**Privileged Clients** from the My Work frame.

Revoking root privileges from a client

This topic describes how to revoke root privileges from a client.

You must Operator or Administrator privileges to perform this task from the SAN File System console. You must Administrator privileges to perform this task from the administrative command-line interface.

1. Click **Manage Servers and Clients**→**Client Sessions** from the My Work frame.
2. Select one or more client sessions from which you want to revoke root privileges.
3. Click **Revoke Clients Root Privileges** from the drop-down box in the table header.

4. Click **Go**.
5. Click **Revoke Root Privileges** to confirm the operation.

Setting administrative privileges and fileset permissions on UNIX

This topic describes how to set up privileges and permissions to enable clients to access UNIX filesets.

You must have Administrator privileges to perform this task.

New filesets are attached with a false user ID and group ID for security reasons. Also *root squashing* (when a root or Administrator user logs into a client that is not a privileged client) makes it seem as if the user has no access.

1. Set the privileged client list.

```
sfscli> chclusterconfig -privclient client_list
```
2. Ensure that you are logged on to a client with administrator privileges.
3. Change to the directory where the fileset is attached. This is the directory given to the **-path** parameter when the fileset was attached.
4. Run the **ls -l** command to verify that the user ID, the group ID, and the permissions are not valid.
5. Issue the **chown** command at the client prompt to change the owner of the directory.
6. Issue the **chgrp** command at the client prompt to change the group ID of the directory.
7. Issue the **chmod** command at the client prompt to change the permissions on the directory.

Setting administrative privileges and fileset permissions on Windows

This topic describes how to set up privileges and permissions to enable Windows clients to access filesets.

You must have Administrator privileges to perform this task.

New filesets are attached with false user ID and group ID for security reasons. Also *root squashing* (when a root or Administrator user logs into a client that is not a privileged client) makes it seem as if the user has no access.

1. Set the privileged client list.

```
sfscli> chclusterconfig -privclient client_list
```
2. Ensure that you are logged on as an administrator or a member of the administrator's group.
3. Open Windows Explorer.
4. Select the attached fileset C2.
5. Right-click and select **Properties**.
6. Select the **Security** tab.
7. Click **Advanced**.
8. Select the **Owner** tab.
9. The owner should be S-1-0-0, which is the null security ID.
10. Choose the appropriate owner, usually Administrator or Administrators.

11. Click **Apply**.
12. Click **OK**.
13. Click **OK** to activate the new settings.
14. Select the **Security** tab.
15. Select the permissions.
16. Click **OK**.

Starting a client

This topic describes how to start a client.

The steps to start the client are specific to the operating system running on the client machine.

Starting an AIX client

This topic provides information for starting an AIX client.

You can use the **setupstclient** command or a series of manual steps to start an AIX client. The client driver is automatically started by starting the operating system.

1. If you are using the **setupstclient** command:
 - a. Type **setupstclient** and press Enter to start the setup guide.
 - b. Enter the appropriate information, when prompted, to set up the virtual client.
2. To manually start the client:
 - a. Create a directory to use as the mount point (for example, mkdir /mnt/sansfs).
 - b. Change to the *base*/client/bin directory, where *base* is the base directory (for example, chdir /usr/tank/client/bin).
 - c. Type **stfsdriver -load <PATH to kernel extension> /stfs** and press Enter to load the kernel extension into AIX.
 - d. Type **./stfsclient -create [client_name] server_IP_address[:port] -converter 8859-1 [-device=kmnamekernel_extension]** and press Enter to create an instance of the SAN File System client, called *virtual client*, where

client_name

The name of the SAN File System client. The default is the host name of the client machine.

server_IP_address

The IP address of the metadata server machine.

port The port of the metadata server machine. The default port is 1700.

kernel_extension

The path to kernel extension, as previously specified to the stfsdriver command, or kernel extension identifier that was returned by the stfsdriver command.

- e. Type **./stfsmount client_name mount_point** and press Enter to mount the specified file system, where

client_name

The name of the client that you specified in step 2d.

mount_point

The directory that you specified in step 2a.

Starting a Linux client

This topic provides information for starting a Linux client.

You can use the **setupstclient** command or a series of manual steps to start a Linux client. The client driver is automatically started by starting the operating system.

1. If you are using the **setupstclient** command:
 - a. Type **setupstclient** and press Enter to start the setup guide.
 - b. Enter the appropriate information, when prompted, to set up the virtual client.
2. To manually start the client:
 - a. Create a directory to use as the mount point (for example, mkdir /mnt/sansfs).
 - b. Change to the *base*/client/bin directory, where *base* is the base directory (for example, chdir /usr/tank/client/bin).
 - c. Type **insmod *driver_file_name*** and press Enter to load the kernel extension into Linux.
 - d. Type **./stfsclient -create [*client_name*] *server_IP_address*[:*port*] -converter 8859-1 [-devices=*pat=pattern*]** and press Enter to create an instance of the SAN File System client, called *virtual client*, where

client_name

The name of the SAN File System client. The default is the host name of the client machine.

server_IP_address

The IP address of the metadata server machine.

port The port of the metadata server machine. The default port is 1700.

-devices=*pat=pattern*

The candidate list is made up of those devices that have device-special files whose file name matches the specified pattern. The default is **-devices=*pat=/dev/sd*[a-z]***, where * represents any alphabetic characters (a-z).

- e. Type **./stfsmount *client_name mount_point*** and press Enter to mount the specified file system, where

client_name

The name of the client that you specified in step 2d.

mount_point

The directory that you specified in step 2a.

Starting a Solaris client

This topic provides information for starting a Solaris client.

You can use the Solaris mount command to start a Solaris client. You use the **setupstclient** only at initial installation of the client.

Starting a Windows client

This topic provides information for starting a Windows-based client.

In a typical setup, the Windows-based client starts automatically when the machine is booted. If your machine is not set up to start automatically, you can start the client manually.

1. Edit the Windows registry by typing **regedt32** and pressing Enter.

2. Navigate to the start key:
`\\HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services \\STFS`
3. Double-click **Start** and change the value to **3** to enable the client from starting automatically when the machine is booted.
4. Click **OK** to close the window.
5. Reboot.

Stopping a client

This topic describes how to stop a client.

The steps to stop the client are specific to the operating system running on the client machine.

Stopping an AIX client

This topic provides information about stopping an AIX client.

You can use the **rmstclient** command or a series of manual steps to stop the AIX client. The client driver is automatically stopped by stopping the operating system.

1. If you are using the **rmstclient** command:
 - a. Type **rmstclient** and press Enter to unmount SAN File System, remove the SAN File System virtual client, and unload the SAN File System driver from the client system.
2. If you are manually stopping the client:
 - a. Type **stfsclient -query -kmname <PATH to kernel extension or kmid that was returned by stfsdriver command>** and press Enter to display a list of installed clients.
 - b. Type **stfsumount *mount_point*** and press Enter to unmount the global namespace, where
mount_point
 The directory that you created to mount to the global namespace.
 - c. Type **./stfsclient -destroy *client_name* -kmname <PATH to kernel extension that was returned from stfsdriver command>** and press Enter to remove the client instance, where
client_name
 The name of the SAN File System client. The default is the host name of the client machine.
client_IP_address
 The IP address of the client machine.
 - d. Type **stfsdriver -unload ./stfs** and press Enter to unload the kernel module.

Stopping a Linux client

This topic provides information about stopping a Linux client.

1. If you are using the **rmstclient** command:
 - a. Type **rmstclient** and press Enter to unmount SAN File System, remove the SAN File System virtual client, and unload the SAN File System driver from the client system.
2. If you are manually stopping the client:
 - a. Use the Linux **umount *mount_point*** command to unmount the global namespace.

- b. Type `/stfscient -destroy client_name` and press Enter to remove the client instance.
- c. Type `rmmod stfs` and press Enter to unload the kernel extension and destroy the instance of the file-system driver.

Stopping a Solaris client

This topic provides information for stopping a Solaris client.

You can use the Solaris `umount` command to stop a Solaris client. You use the `rmstclient` only when you want to uninstall the client entirely.

Stopping a Windows client

This topic provides information about stopping a Windows client.

You cannot stop the Windows clients without shutting down the entire system. If the automatic-restart setting is enabled, it will restart automatically when the system is restarted; however, you can disable the automatic-restart setting.

1. Edit the Windows registry by typing `regedt32` and pressing Enter.
2. Navigate to the start key:
`\\HKEY_LOCAL_MACHINE\\SYSTEM\\CurrentControlSet\\Services \\STFS`
3. Double-click **Start** and change the value to **2** to disable the client from starting automatically when the machine is booted.
4. Click **OK** to close the window.
5. Reboot.

Viewing client-session details

This topic describes how to display the details for a client session.

1. Click **Manage Servers and Clients**→**Client Sessions** from the My Work frame.
2. Select a client session.
3. Click **Details** from the drop-down box in the table header.
4. Click **Go**.

Viewing client-session statistics

This topic describes how to view client-session statistics.

1. Click **Monitor System**→**Statistics** from the My Work frame.
2. Click the **Client Sessions** tab.
3. Click **OK** to view the statistics.
4. Click **Close** to close the Statistics panel.

Managing the cluster

A SAN File System cluster is a set of metadata servers and engines with one metadata server running on each engine. The servers in a cluster communicate with each other and with SAN File System clients over your existing IP network. A cluster provides a single point of control for administrative and service operations.

You can perform the following cluster tasks from the SAN File System console:

Changing active cluster states

This topic describes how to change the active state of the cluster.

You must have Operator or Administrator privileges to perform this task.

1. Click **Manage Servers and Clients**→**Cluster** from the My Work frame.
2. Click **Change State** from the drop-down box in the table header.

Note: A cluster cannot be changed from an inactive (Not Running, Forming) or unknown state, to an active state (Online, Partly Quiescent, Fully Quiescent, and Offline); or from an active to an inactive state using the SAN File System console.

3. Click **Go**.
4. Select a new active state for the cluster.
5. Click **OK** to confirm the state change.

Configuring the cluster language

This topic describes how to configure the language in which the metadata server runs for the cluster.

You must have Administrator privileges to perform this task.

Requirement: If you change the cluster language setting, you must restart the cluster for the change to take affect in the cluster logs. For the language change to take affect in the administrative agent logs, you must restart the administrative agent using the **stopCimom** and **startCimom** commands or reboot the engine.

1. Click **Manage Servers and Clients**→**Cluster** from the My Work frame.
2. Click **Properties** from the drop-down box in the table header.
3. Click **Go**.
4. Click the **Localization** tab.
5. Select the language from the **Language** drop-down box.
6. Click **OK** to enable the new setting.

Configuring cluster tuning

This topic describes how to configure tuning for the SAN File System cluster.

You must have Administrator privileges to perform this task.

1. Click **Manage Servers and Clients**→**Cluster** from the My Work frame.
2. Click **Properties** from the drop-down box in the table header.
3. Click **Go**.
4. Click the **Tuning** tab.
5. Select the check box to enable **Pool Space Reclamation**, or clear the check box to disable this action.
6. If enabled, enter an interval from 1 to 1440 minutes.
7. Click **OK** to enable the new settings.

Listing cluster configuration

This topic describes how to list the SAN File System cluster configuration.

To view a listing of the SAN File System cluster configuration, click **Manage Servers and Clients**→**Cluster** from the My Work frame.

Starting the cluster

This topic describes how to start the SAN File System cluster. You can start the cluster in an offline or online state.

You must have Operator or Administrator privileges to perform this task.

If the cluster is not already running (that is, specifically in the “not running” state), you can start it from the SAN File System console.

1. Click **Manage Servers and Clients**→**Cluster** from the My Work frame.
2. Click **Start Online** or **Start Offline** from the drop-down box in the table header.
3. Click **Go**.

Stopping the cluster

This topic describes how to stop the SAN File System cluster.

You must have Operator or Administrator privileges to perform this task.

If the cluster is not already in the “not running” state, you can stop it from the SAN File System console.

1. Click **Manage Servers and Clients** → **Cluster** from the My Work frame.
2. Click **Stop** from the drop-down box in the table header.
3. Click **Go**.
4. Verify that you have selected the servers that you want to stop, and click **OK** to confirm the cluster stop operation.

Upgrading the cluster

This topic describes how to upgrade the currently operational cluster software version.

You must have Operator or Administrator privileges to perform this task. Additionally, the cluster upgrade can occur only after each individual metadata server has been upgraded to the same new software version.

1. Click **Manage Servers and Clients** → **Cluster** from the My Work frame.
2. Click **Upgrade** from the drop-down box in the table header.
3. Click **Go**.
4. Click **OK** to confirm the upgrade.

Viewing cluster details

This topic describes how to view details for the SAN File System cluster.

1. Click **Manage Servers and Clients** → **Cluster** from the My Work frame.
2. Click **Properties** from the drop-down box in the table header.
3. Click **Go**.
4. Click the **Details** tab.

Viewing cluster statistics

This topic describes how to view cluster statistics.

1. Click **Monitor System** → **Statistics** from the My Work frame.
2. Click the **Cluster** tab.
3. Click **OK** to view the statistics.
4. Click **Close** to close the Statistics panel.

Viewing the cluster software version

This topic describes how to view the operational and pending cluster software versions.

1. Click **Manage Servers and Clients** → **Cluster** from the My Work frame.
2. View the current software version in the **Software Version** field.
3. View the next version that the software will move to in the **Committed Software Version** field.

Managing disaster recovery

The SAN File System console enables you to create and delete files to assist in disaster recovery. In addition, there are several disaster recovery tasks that can be performed from the Administrative command-line interface.

Several of the restoring tasks depend on having first executed certain backup tasks. The method of restoration will depend a great deal upon whether you chose the LUN or API method of backup.

Creating a recovery file

This topic describes how to create a file for disaster recovery.

You must have Operator or Administrator privileges to perform this task.

1. Click **Maintain System**→**Disaster Recovery** from the My Work frame.
2. Click **Create** from the drop-down box in the table header.
3. Click **Go**.
4. Select the check box to **Create a new recovery file** or select the check box for a **Forced Create**, which will overwrite an existing file.

Attention: When you overwrite an existing file, metadata recovery of items from that file may not be possible.

5. Type a name for the newly created file, or select the **Existing Recovery File** to overwrite from the drop-down menu.
6. Click **OK** to confirm the creation of the new file or to overwrite the existing one.
7. Click **Maintain System**→**Disaster Recovery** from the My Work frame to verify that the recovery file was created.

Creating recovery scripts

This topic describes how to create recovery scripts from a recovery file. The recovery file and these scripts are required for restoring SAN File System metadata.

You must have Backup, Operator, or Administrator privileges to perform this task.

1. Use the **builddrscript** command and specify the name of the recovery file, which you created using the **Maintain System** task in SAN File System console or the **mkdrfile** command in the administrative command-line interface:

```
sfsccli builddrscript recovery-file-name
```
2. The **builddrscript** command stores the recovery scripts in the /usr/tank/server/DR directory on the master metadata server. See “Restoring SAN File System metadata” on page 83 for information about editing and running these scripts to restore SAN File System metadata.

Deleting a recovery file

This topic describes how to delete a disaster recovery file.

You must have Operator or Administrator privileges to perform this task.

1. Click **Maintain System**→**Disaster Recovery** from the My Work frame.
2. Select a recovery file for deletion.
3. Click **Delete** from the drop-down box in the table header.
4. Click **Go**.

Attention: When you delete an existing recovery file, metadata recovery of items from that file may not be possible.

5. Click **Delete** to confirm the file deletion.

Listing recovery files

This topic describes how to display a list of all recovery files.

To display a list of all recovery files, click **Maintain System**→**Disaster Recovery** from the My Work frame.

Restoring the master console

This topic explains how to restore the hardware and the operating system for the master console.

1. Determine if the hardware for the master console is working properly. If so, review information about recovering the hard drives (if necessary) as well as recovering the software.
2. If the hardware for the master console is not working properly,
 - a. Refer to your server documentation to resolve problems with the hardware.
 - b. Refer to the *Planning, Installation and Configuration Guide* for information about installing the master console.

Restoring the engine hardware and operating system

This topic explains how to restore the hardware and the operating system for an engine.

1. Verify that there is no damage to the hardware and that the engine boots properly. If you suspect a problem with any of the hardware components, troubleshoot an engine to resolve the problem.
2. Verify that there is no damage to the master console and that it boots properly.
 - a. If you suspect a problem with any of the hardware components in the master console, refer to your server documentation to resolve those problems.
 - b. If you suspect a problem with the software or the hard disk drive, troubleshoot the master console to resolve the problem.

3. From the master console, point the Web browser to the URL of the RSA II adapter on the engine and access the RSA II adapter to set up a remote console to the engine. This interface allows you to use the master console as your display and keyboard for the engine.

Note: Instead of using the RSA II Web interface from the master console, you can directly attach a keyboard and display to the engine. However, make sure that you attach the display to the VGA port of the RSA II card on the engine and not to the video port on the engine itself.

4. Determine if the boot drives for each engine hosting a metadata server still have an intact SAN File System configuration and executable files (undamaged).
5. If there are corrupt or damaged configuration and executable files, attempt to recover the damaged files from the mirrored boot drive. If you cannot recover the damaged files from the mirrored boot drive:
 - a. Load the Disaster Recovery CD into the CD-ROM drive on the engine.
 - b. Reboot the engine using one of the following methods:
 - 1) Open a bash shell prompt and enter **init 6**.
 - 2) Press the Reset button on the front panel of the engine.
 - 3) Power off the engine and then power it back on.
 - c. When you receive a warning prompt that the entire hard drive will be overwritten, respond by entering **y**.
 - d. After the operating system has been reloaded, the engine will eject the Disaster Recovery CD and automatically reboot.

Restoring SAN connectivity

This topic explains how to restore connectivity between the SAN File System and the SAN.

1. If the system was backed up using the LUN method, perform these steps on each engine in the SAN File System cluster to restore SAN connectivity:
 - a. Verify that the engines hosting the metadata servers are connected to the SAN in the same configuration that existed at the point of the last backup operation (make sure the metadata servers can see the same LUNs that existed prior to the unexpected outage).
 - b. If the LUN mapping has changed, use the device management tools for the storage subsystem or management tools for the SAN to recreate the old LUN map. After creating the old LUN map, reboot the metadata server so that the changes to the LUN map are visible to the metadata server.
 - c. If LUN contents were lost or corrupted, use the copy services facility of the storage subsystem to restore all LUN data (both metadata and user file data).
2. If the system was backed up using the API method, perform these steps on each engine in the SAN File System cluster to restore SAN connectivity.
 - a. Verify that the engines hosting the metadata servers are connected to the SAN in the same configuration that existed at the point of the last backup operation (make sure the metadata servers can see the same LUNs that existed prior to the unexpected outage).
 - b. If the LUN mapping has changed, use the device management tools for the storage subsystem or management tools for the SAN to recreate the old

LUN map. You can also choose to restore data onto a new LUN map. However, if you do so, you will have to manually run some of the steps used to restore metadata.

Restoring SAN File System software

This topic explains how to restore the metadata server and administrative server software on an engine.

1. Reinstall the software for the metadata server.
 - a. Make sure that you are logged into the engine as root.
 - b. From a shell prompt on the engine, change to the directory where the metadata server software package is installed.

```
cd /usr/tank/packages
```
 - c. Install the metadata server software package using the following command:

```
bash# rpm -ivh metadata_server_package_name.rpm
```
 - d. Install the Qlogic software using the following command:

```
bash# rpm -i snia_qlogic_hba*.rpm
```
2. Reinstall the software package for the administrative server.
 - a. Install the administrative server software package using the following command:

```
bash# rpm -ivh administrative_server_package_name.rpm
```

Restoring SAN File System cluster configuration

This topic explains how to restore the configuration for the SAN File System cluster.

1. If the system was backed up using the LUN method, and the entire cluster is down, perform these steps to restore the cluster configuration information:
 - a. If you have previously saved the configuration files to another location, copy these files onto the boot drive for the engine.
 - 1) Copy Tank.Bootstrap to /usr/tank/server/config.
 - 2) Copy Tank.Config to /usr/tank/server/config.
 - Note:** If you have saved any other administrative configuration files, you can reference them when restoring the SAN File System metadata configuration.
 - b. If the cluster bootstrap file, Tank.Bootstrap, is corrupted or missing, you can attempt to recreate the contents of that file using information from the metadata LUNs:
 - 1) Use the /usr/tank/server/bin/tank lsdisklabel -device command to find the master volume. If you cannot remember which device is your master volume, this is an iterative process of searching all suspected master volume devices until the command indicates you have found a valid master volume.
 - 2) Use the /usr/tank/server/bin/tank extractbootrecord command to regenerate Tank.Bootstrap from the master volume.
 - 3) Use the /usr/tank/server/bin/tank resetcluster command to reinitialize the master volume for subsequent rebuilding of the cluster configuration.
 - 4) Use the /usr/tank/server/bin/tank addserver command for all subordinate metadata server engines to recreate the cluster definition.

2. If the system was backed up using the LUN method, and only the master metadata server is down, perform these steps to restore the cluster configuration information:
 - a. Change the master metadata server to one of the subordinate metadata servers:
 - 1) Run `/usr/tank/server/bin/tank lsserver` to verify that the cluster does not have a master metadata server.
 - 2) Run `/usr/tank/server/bin/tank lsengine` to verify that the engine hosting the master metadata server is shut down.
 - 3) Run `/usr/tank/server/bin/tank lsserver` to verify that the metadata server to which you are going to assign the role of master is up and running.
 - 4) Access the subordinate metadata server that you want to set as the new master metadata server.
 - 5) Run the `/usr/tank/server/bin/tank setmaster` command to set it as the new master metadata server.
 - 6) If the previous master metadata server was managing any filesets, reassign those filesets to other metadata servers in the cluster.
 - b. If you have previously saved the configuration files to another location, copy these files onto the boot drive for the engine.
 - 1) Copy `Tank.Bootstrap` to `/usr/tank/server/config`.
 - 2) Copy `Tank.Config` to `/usr/tank/server/config`.

Note: If you have saved any other administrative configuration files, you can reference them when restoring the SAN File System metadata configuration.

 - c. If the cluster bootstrap file, `Tank.Bootstrap`, is corrupted or missing, you can attempt to recreate the contents of that file using information from the metadata LUNs:
 - 1) Use the `/usr/tank/server/bin/tank lsdisklabel -device` command to find the master volume. If you cannot remember which device is your master volume, this is an iterative process of searching all suspected master volume devices until the command indicates you have found a valid master volume.
 - 2) Use the `/usr/tank/server/bin/tank extractbootrecord` command to regenerate `Tank.Bootstrap` from the master volume.
 - 3) Use the `/usr/tank/server/bin/tank resetcluster` command to reinitialize the master volume for subsequent rebuilding of the cluster configuration.
 - 4) Use the `/usr/tank/server/bin/tank addserver` command to add the original master metadata server to the cluster.
 - d. Repeat steps 2a1 through 2a6 to change the master metadata server back to the original metadata server.
 - e. Use the `/usr/tank/server/bin/tank resetcluster` command to reinitialize the master volume for subsequent rebuilding of the cluster configuration.
3. If the system was backed up using the API method, perform these steps to restore the cluster configuration information:
 - a. If you have previously saved the configuration files to another location, copy these files onto the boot drive for the engine.
 - 1) Copy `Tank.Bootstrap` to `/usr/tank/server/config`.
 - 2) Copy `Tank.Config` to `/usr/tank/server/config`.

- Note:** If you have saved any other administrative configuration files, you can reference them when restoring the SAN File System metadata configuration.
- b. If you suspect that the metadata LUNs are corrupted, you can perform these steps to recreate the cluster definition:
 - 1) Delete all Tank.Bootstrap and Tank.Config files from your metadata server engines.
 - 2) Start the `/usr/tank/server/bin/tank` binary on your master metadata server with the `install` option rather than *normal* option.

Attention: The existing metadata data server information will be overwritten.

This will create new Tank.Bootstrap and Tank.Config files on your metadata server master. Be sure to specify the same cluster name that was used prior to the disaster:
 - 3) Now start the master metadata server with `/usr/tank/server/bin/tank normal` command.
 - 4) Use the `addserver` command to add all subordinate metadata server engines. This will create new Tank.Bootstrap and Tank.Config files on the subordinates.

Restoring SAN File System metadata

This topic explains how to restore the metadata for the SAN File System cluster.

1. Verify that all metadata servers in the cluster are online and that the cluster is running.


```
sfscli lserver -state online
```
2. Copy the system-metadata disaster-recovery file (and the scripts) that you had previously backed up to `/usr/tank/server/DR` on the master metadata server.
3. Use the TankSysCLI.auto script:
 - a. Edit the script TankSysCLI.auto for information about how the script is used and any changes that might need to be made to the script.


```
#####
# CLI Commands to create Storage Pools, Filesets, Service Classes and
# Policy Sets.
# These commands need NO manual intervention.
#####
```
 - b. Run the script TankSysCLI.auto.


```
sfscli -script /usr/tank/server/DR/TankSysCLI.auto
```
 - c. If any errors occur while running the script, ensure that you resolve those errors before continuing.
4. Use the TankSysCLI.volume script:
 - a. Edit `/usr/tank/server/DR/TankSysCLI.volume` and modify it to match your current SAN settings. It also contains usage information as well as information about any changes that might need to be made to the script.


```
#####
# CLI Commands to add Volumes to Storage pools.
# These commands need manual intervention.
# The device names were as they appeared during backup.
# Please make sure that the device names appearing here actually
# exist and have correct sizes and if not edit the device names to
# correct values.
# The System MASTER volume has to be specified in tank install command
# and therefore has no corresponding CLI.
# The other System Volumes can either be specified in tank install
```

- ```
command, or, added using the CLI command, which appears inside comments
forthis reason.
#####
```
- b. Run the script TankSysCLI.volume.

```
sfscli -script /usr/tank/server/DR/TankSysCLI.volume
```
  - c. If any errors occur while running the script, ensure that you resolve those errors before continuing.
5. Use the TankSysCLI.attachpoint script:
    - a. Edit /usr/tank/server/DR/TankSysCLI.attachpoint to verify the settings. It also contains usage information as well as information about any changes that may need to be made to the script.

```
#####
CLI Commands to attach filesets.
These commands need manual intervention.
All the "mkdir" and "attachfileset" commands should be run in the
order given.
The "mkdir" command should be run on a client to recreate the directory
path before running the following attachfileset CLI commands.
#####
```
    - b. If all filesets are attached only to the root directories of other filesets, run the script TankSysCLI.attachpoint.

```
sfscli -script /usr/tank/server/DR/TankSysCLI.attachpoint
```

**Note:** If you have any filesets attached to directories, you must reattach them manually.
    - c. If any errors occur while running the script, ensure that you resolve those errors before continuing.
  6. Grant privileges to those clients that require root or Administrator access to SAN File System using the **chclusterconfig -privclient** command.

## Restoring SAN File System clients

This topic explains how to restore SAN File System clients.

SAN File System clients are access points to the SAN File System. Therefore, clients are not backed up. To restore a SAN File System clients, you can perform the normal client installation procedure, which is described in the *Planning, Installation, and Configuration Guide*.

## Restoring SAN File System user data

This topic explains how to restore SAN File System user data.

1. From a client, mount the SAN File System at its usual mount point. The top of the subdirectory tree (the portion of the subdirectory tree that consists of the fileset names) should be visible from the client.
2. Restore the files onto that mount point. Follow the procedures for the backup and recovery application to back up the files.
3. If you followed the guidelines in the *Planning, Installation, and Configuration Guide* for backup and recovery, restore files to the Windows filesets from a Windows client and restore files to the UNIX filesets from a UNIX-based client.

---

## Managing engines

Within SAN File System, an engine is the hardware (which is based on the IBM xSeries platform) on which a metadata server and an administrative server run.

The administrative infrastructure on each engine allows an administrator to monitor and control SAN File System from a standard Web browser or an administrative command-line interface. The two major components of the infrastructure are an administrative agent, which provides access to administrative operations, and a Web server that is bundled with the console services and servlets that render HTML for the administrative browsers.

You can perform the following tasks on SAN File System engines:

## Accessing an engine through SSH

This topic describes how to remotely access an engine using SSH from the master console.

Before accessing an engine using SSH, the following requirements must be met:

- A SAN File System administrative user account must be set up for use in signing on to the SAN File System console.
  - A UNIX-based operating system account must be set up on the engine to be accessed for use in signing on to the SSH session.
  - If the service representative is accessing the master console remotely:
    - You must have previously initiated a VPN connection with the service representative.
    - The service representative must have established the VPN connection.
1. From the master console, use one of these methods to access the engine:
    - a. Open a shell prompt and type `putty.exe -ssh engine_IP_address`, where *engine\_IP\_address* is the IP address of the engine to be accessed.

**Note:** If you used SSH to establish a remote session with the master console, type this command from that session to establish an SSH session between the master console and the engine.

- b. Click **Start>Programs>PuTTY>PuTTY**.
    - 1) Type the IP address of the engine to be accessed.
    - 2) Select SSH as the protocol.
    - 3) Click Open.
2. After the session is established, log in using a UNIX-based user ID and password.

After connecting to the engine, you can perform these activities:

- Access the SAN File System administrative command-line interface (CLI) to run SAN File System commands. These commands provide the ability to manage engines, metadata servers, and administrative servers.
- Access operating-system commands to enable or disable tracing, obtain dumps, and stop or start applications.

## Accessing the RSA II adapter

This task describes how to access the RSA adapter that is installed on each engine in the cluster.

Before accessing the RSA adapter for an engine, the following requirements must be met:

- An RSA user account must be set up for use in accessing the RSA II Web interface on the engine.

- If the service representative is accessing the master console remotely, the following conditions must be met:
  - You must have previously initiated a VPN connection with the service representative.
  - The service representative must have established the VPN connection and used a remote display emulation package, such as VNC, to remotely view the desktop of the master console.
- 1. Open a browser from the remote console on the master console and type the URL for an RSA adapter:  
`http://RSA_II_web_address/`
- 2. Log on to the RSA II interface using a valid RSA user name and password.

From the left navigation pane, you can choose to perform tasks such as:

- Stopping and restarting the engine
- Viewing vital product data for the engine
- Accessing the BIOS and firmware for the engine
- Updating the firmware for the RSA card
- Accessing the RSA adapters of other engines in the SAN File System cluster

**Note:** If, when you view the System Health Summary page, all components are listed as unavailable, make sure the PCI-riser card assembly is firmly seated within the engine.

You can also start a remote video console that can be used to redirect the engine console to the master console. In addition, you can use the remote control functionality to assign the CD-ROM or diskette drive from the master console to be used by the engine.

**Note:** To use the remote control functionality of the RSA adapter, you must be signed on with an ID defined in the RSA II card that has read/write access to the RSA II adapter.

For more information about using the RSA card, see the *Remote Supervisor Adapter II User's Guide*, which is available from this Web site (search for Remote Supervisor Adapter II from the Search Technical Support link):

[www.ibm.com/storage/support/](http://www.ibm.com/storage/support/)

## Collecting diagnostic data on engines

This topic describes how to collect diagnostic data for engines using the one-button data collection function from the SAN File System console.

You must have Administrator or Operator privileges to perform this task.

1. Click **Maintain System**→**Collect Diagnostic Data** from the My Work frame.
2. Select the engine or engines on which you would like to collect diagnostics.
3. Click **OK**.

After the data is collected, a progress panel will appear that displays a listing of all the engines on which diagnostics were created, and the file path where the data is stored.

## Listing engines

This topic describes how to display a list of all engines.

To display a list of all engines, click **Maintain System → Engines** from the My Work frame.

## Powering on the engine

This topic describes how to power off an engine.

You must have Administrator or Operator privileges to perform this task.

After you power off the engine, wait at least five seconds before you power on the engine again.

1. Click **Maintain System → Engines** from the My Work frame.
2. Select an engine.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Power** tab.
6. Click **Change Power State**.
7. Select the **Power Off**, **Forced Power Off**, or **Scheduled Power On or Off** radio button.
8. Fill in any appropriate fields.
9. Click **OK** to power off the engine.

## Powering on the engine

This topic describes how to power on an engine.

- You must have Administrator or Operator privileges to perform this task.
- Verify that all external devices are powered on and that the engine power cords are plugged into the power source.

If the engine is powered on and a power failure occurs, the engine restarts automatically when power is restored.

While the engine is in the process of powering on, the power-on LED on the front of the engine is lit.

1. Click **Maintain System → Engines** from the My Work frame.
2. Select an engine.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Power** tab.
6. Click **Change Power State**.
7. Select the **Power On** or **Scheduled Power On or Off** radio button.
8. Fill in any appropriate fields.
9. Click **OK** to power on the engine.

## Restarting the engine

This topic describes how to reboot an engine.

You must have Administrator or Operator privileges to perform this task.



1. Click **Maintain System** → **Engines** from the My Work frame.
2. Select an engine.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Power** tab.
6. Click **Change Power State**.
7. Select the **Power Cycle** radio button.
8. Click **OK** to reboot the engine.

## Starting the Configuration/Setup Utility

This topic describes the Configuration/Setup Utility and how to start it.

1. The Configuration/Setup Utility program is a menu-driven utility that is part of the BIOS code that comes with the Model 1RX. You can use it to perform these functions:
  - Configure serial port assignments.
  - Change the drive startup sequence.
  - Enable USB keyboard and mouse support.
  - Resolve configuration conflicts.
  - Set the date and time.
  - Set passwords.
2. Perform the following steps to start the Configuration/Setup Utility program:
  - a. Power ON the engine and watch the monitor screen.
  - b. When the message Press F1 for Configuration/Setup appears, press F1.

**Note:** To run the Configuration/Setup Utility, you must use the highest-level password available for the engine when you power it on. For example, if an administrative password is set for the engine, you must use it when you power on the engine instead of using the power-on password.

If you do not use the administrative password, you will have access to only a limited set of Configuration/Setup Utility functions.

- c. Follow the instructions that appear on the screen.

## Viewing engine fan status

This topic describes how to display a summary of a specific for a engine.

1. Click **Maintain System** → **Engines** from the My Work frame.
2. Select an engine.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Fans** tab.

## Viewing engine power status

This topic describes how to display the power status for a specific engine.

1. Click **Maintain System**→**Engines** from the My Work frame.
2. Select an engine.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.

5. Click the **Power** tab.

## Viewing engine statistics

This topic describes how to view engine statistics.

1. Click **Monitor System**→**Statistics** from the My Work frame.
2. Select the **Engine** tab.
3. Click **OK** to view the statistics.
4. Click **Close** to close the Statistics panel.

## Viewing the engine status summary

This topic describes how to display a status summary of a specific for a engine.

1. Click **Maintain System**→**Engines** from the My Work frame.
2. Select an engine.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Summary** tab.

## Viewing engine temperatures

This topic describes how to display the temperatures for a specific for a engine.

1. Click **Maintain System**→**Engines** from the My Work frame.
2. Select an engine.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Temperatures** tab.

## Viewing engine timeouts

This topic describes how to display the timeout settings for a specific for a engine.

1. Click **Maintain System**→**Engines** from the My Work frame.
2. Select an engine.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Timeouts** tab.

## Viewing engine voltage status

This topic describes how to display the voltage status for a specific for a engine.

1. Click **Maintain System**→**Engines** from the My Work frame.
2. Select an engine.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Voltages** tab.

## Viewing vital engine data

This topic describes how to display the vital data for a specific engine.

1. Click **Maintain System**→**Engines** from the My Work frame.
2. Select an engine.
3. Click **Properties** from the drop-down box in the table header.

4. Click **Go**.
5. Click the **Vital Engine Data** tab.

---

## Managing filesets

A fileset (except for the global fileset) is a subset of the entire SAN File System global namespace. It serves as the unit of workload for metadata servers and is also the unit that an administrator specifies to create FlashCopy images that are used in backup procedures.

From a client perspective, a fileset appears to be a regular directory. Users and applications on SAN File System clients can create objects, such as directories and files, within the fileset.

### Attaching a fileset

This topic describes how to attach a fileset to the global namespace.

You must have Administrator privileges to perform this task.

The attach point (*existing directory/new directory*) appears as a drive and directory on Windows clients or as a file system on UNIX-based clients. It is recommended that you map the name of the directory to its associated fileset by using the same name as the fileset plus an additional suffix to distinguish it as the directory and not the fileset. For example, if you want to attach the fileset named *work* to the root of the global namespace, you would specify “sanfs” for the *existing directory* and specify “work\_ap” for the *new directory*. On a Windows client, the “work\_ap” directory would appear under the “sanfs” drive.

1. Click **Manage Filing→Filesets** from the My Work frame.
2. Select the fileset that you want to attach.
3. Click **Attach** from the drop-down box in the table header.
4. Click **Go**.
5. Fill in the appropriate fields.
6. Click **OK** to attach the fileset.

### Changing fileset settings

This topic describes how to modify the general settings of a specific fileset.

You must have Administrator privileges to perform this task.

1. Click **Manage Filing→Filesets** from the My Work frame.
2. Select a fileset.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **General Settings** tab to change the fileset name, description and metadata server, or click the **Quota Options** tab to change the quota and alert settings.
6. Edit the appropriate fields.
7. Click **OK** to save the new settings.

## Creating a fileset

This topic describes how to create a fileset.

You must have Administrator privileges to perform this task.

Although SAN File System allows you to create nested filesets, avoid creating nested filesets for these reasons:

- You cannot access a child fileset if the parent fileset is unavailable.
  - You cannot revert a FlashCopy image of a fileset that has nested filesets attached to it (for example, if /fileset1/fileset2, you must detach fileset2, revert to a FlashCopy image of fileset1, and then reattach fileset2).
  - You cannot make FlashCopy images of multiple filesets (whether the filesets are siblings or nested) in one operation.
  - Backup and restore of filesets that are used primarily by one platform should be performed on the primary platform. If the parent and nested filesets are each used primarily by different platforms, you cannot effectively perform backup and recovery from one platform.
1. Click **Manage Filing→Create a Fileset** from the My Work frame.
  2. Fill in the appropriate fields. **Attention:** Clicking the **Toggle Help Assistant** button after entering values in the fields will clear the fields.
  3. Click **OK** to create the fileset.
  4. Click **Manage Filing→Filesets** from the My Work frame to verify that the fileset was created.

## Deleting a fileset

This topic describes how to delete a fileset.

You must have Administrator privileges to perform this task.

1. Click **Manage Filing→Filesets** from the My Work frame.
2. Select the fileset that you want to delete.
3. Click **Delete** from the drop-down box in the table header.
4. Click **Go**.
5. Select the deletion method.
6. Click **OK**.
7. Click **OK** to confirm the deletion.

## Detaching a fileset

This topic describes how to detach a fileset from the global namespace.

You must have Administrator privileges to perform this task.

The fileset that you want to detach must not have subfilesets. You must detach any subfilesets before detaching the desired fileset.

The directory to which the fileset was attached is deleted after the fileset is detached.

1. Click **Manage Filing→Filesets** from the My Work frame.
2. Select the fileset that you want to detach.
3. Click **Detach** from the drop-down box in the table header.
4. Click **Go**.

5. Select the appropriate detachment method.
6. Click **OK** to confirm the detachment.

## Listing filesets

This topic describes how to display a list of all filesets.

To display a list of all filesets, click **Manage Filing→Filesets** from the My Work frame.

## Reassigning a fileset to a different metadata server

This topic describes how to reassign a fileset to a different metadata server.

You must have Administrator privileges to perform this task.

1. Click **Managing Filing→Filesets** from the My Work frame.
2. Select the fileset to reassign.
3. Click **Properties** from the drop-down box in the table.
4. Click **Go**.
5. Click the **General Settings** tab.
6. Select a metadata server from the **Server** drop-down box.
7. Click **Apply**.

## Viewing fileset details

This topic describes how to view details for a specific fileset.

1. Click **Manage Filing→Filesets** from the My Work frame.
2. Select a fileset.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Details** tab.

## Viewing fileset placement information

This topic describes how to view placement information for a fileset, such as its storage pool, whether the storage pool is currently in use, and the number of policy rules that associate the storage pool with the fileset.

You must have Administrator privileges to perform this task.

1. Click **Manage Filing→Filesets** from the My Work frame.
2. Select the fileset from the table.
3. Click **Details of the File Placements in Pools** from the drop-down box in the table header.
4. Click **Go**.

## Viewing fileset settings

This topic describes how to view settings for a specific fileset.

1. Click **Manage Filing→Filesets** from the My Work frame.
2. Select a policy.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **General Settings** tab.

## Viewing fileset statistics

This topic describes how to view fileset statistics.

1. Click **Monitor System**→**Statistics** from the My Work frame.
2. Select the **Fileset** tab.
3. Click **OK** to view the statistics.
4. Click **Close** to close the Statistics panel.

---

## Managing FlashCopy images

A FlashCopy image is a space-efficient, read-only copy of the contents of a fileset in a SAN File System global namespace at a particular point in time. A FlashCopy image can be used with standard backup tools available in your environment to create backup copies of files on tape.

### Creating a FlashCopy image

This topic describes how to create a FlashCopy image of a fileset.

SAN File System provides a wizard to step you through the process of creating a FlashCopy image.

You must have Administrator, Operator, or Backup privileges to perform this task.

1. Click **Maintain System**→**Create FlashCopy Images** from the My Work frame.
2. Follow the steps in the wizard to create a FlashCopy image.
3. Click **Maintain System**→**FlashCopy Images** from the My Work frame and verify that the FlashCopy image was created.

### Deleting a FlashCopy image

This topic describes how to delete a FlashCopy image.

You must have Administrator, Operator, or Backup privileges to perform this task.

1. Click **Maintain System**→**FlashCopy Images** from the My Work frame.
2. Select one or more FlashCopy images to delete.
3. Click **Delete** from the drop-down box in the table header.
4. Click **Go**.
5. Click **OK** to confirm the deletion.

### Listing FlashCopy images

This topic describes how to display a list of all FlashCopy images.

To display a list of all FlashCopy images for a specific fileset, click **Maintain System**→**FlashCopy Images** from the My Work frame.

### Reverting to a previous FlashCopy image

This topic describes how to revert to a previous a FlashCopy image.

**Attention:** When you revert to a FlashCopy image, all FlashCopy images created after the specified FlashCopy image are deleted. The specified FlashCopy image becomes the primary image for the fileset and no longer appears as an image listed in the .flashcopy directory.

You must have Administrator, Operator, or Backup privileges to perform this task.

1. Click **Maintain System**→**FlashCopy Images** from the My Work frame.
2. Select the FlashCopy image that you want the fileset to revert to
3. Click **Revert to** from the drop-down box in the table header.
4. Click **Go**.
5. Click **OK** to revert to specified the FlashCopy image.

## Viewing FlashCopy image details

This topic describes how to view details for a specific FlashCopy image.

You must have Administrator or Operator privileges to perform this task.

1. Click **Maintain System**→**FlashCopy Images** from the My Work frame.
2. Select a FlashCopy image to view.
3. Click **Details** from the drop-down box in the table header.
4. Click **Go**.

---

## Managing metadata servers

In addition to providing metadata to clients and managing locks, metadata servers perform a wide variety of other tasks. They process requests issued by administrators to create and manage filesets, storage pools, volumes, and policy sets, and they enforce the policies defined by administrators to place files in appropriate storage pools and ensure that capacity quotas established for filesets and storage pools are not exceeded.

You can perform the following metadata server tasks:

## Remotely accessing the master console

This task describes how to remotely access the master console.

Before initiating a VPN connection with an IBM representative, the following requirements must be met:

- The master console must have a connection to the Internet.
- A Windows user account for the support representative must be set up on the master console.
- If the support representative has a need to access the SAN File System console or the RSA II Web interface remotely, a remote display emulation package, such as Virtual Networking Computer (VNC) server, must be installed and running on the master console.
- The customer must provide a user ID and password for access.
- A maintenance agreement must be established between you and IBM or the product must be under software warranty.

The master console is used to set up a VPN connection between you and IBM support representatives. You initiate the connection and have the ability to monitor and control the connection.



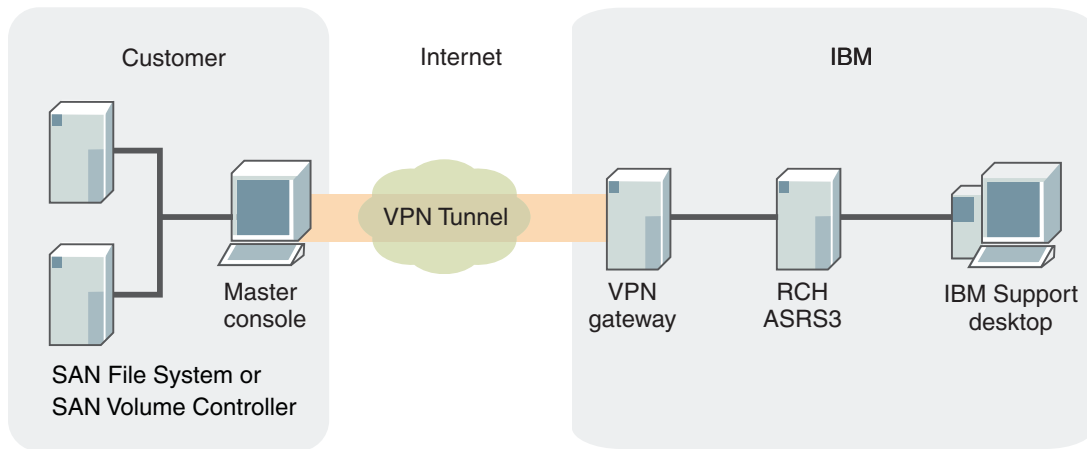


Figure 7. VPN connection between the customer and IBM support representative

1. Log into the master console. You can access the master console directly (using the keyboard, monitor, and mouse) or remotely through another computer on the same LAN.
2. Establish a secure VPN connection from the master console through the VPN gateway to a previously designated VPN server within the IBM intranet. Establish the connection using the IBM connection manager and obtain a connection ID. The IBM connection manager icon is located on the master console desktop.
3. Provide the connection ID to the service representative. Each time you start a VPN session, a unique connection ID is created.
4. The service representative connects to the previously designated VPN server within IBM using either a telnet client or a secure shell (SSH) client, such as PuTTY. The service representative uses the connection ID that you provide to access the active VPN tunnel.
5. The service representative connects to an account on the master console over the VPN connection to access to the master console. The service representative then establishes a second connection to the VPN server. If a remote display emulation package is available, the service representative can use this connection to establish a remote console to the master console.

This connection provides a service representative with the ability to log on to these devices or interfaces:

- Each of the engines in the SAN File System cluster. The support representative can query and control the engines at the operating system level by initiating an SSH session with the engine. This requires that a UNIX-based user account be set up on each of the engines in the cluster.
- The administrative command-line interface. The support representative can query and control the SAN File System metadata servers, and access metadata, log, dump, and configuration data. This requires that a SAN File System administrative user account be set up for the support representative.
- SAN File System clients. The support representative can query and control clients at the operating system level by initiating either an SSH session or a telnet session with the client (if an SSH or telnet application is installed and running on the client). This requires that an operating system user account be set up on each of the clients to which the support representative will need access.
- SAN File System console and RSA II Web interface (if a remote display emulation package is installed and running).

You can monitor all activity performed by the service representative. You can either run a remote desktop package from another machine to observe the master console desktop, view the master console SSH log file to see the results of all activity, or watch directly from the monitor on the master console. In addition, you can disconnect the VPN session at any time.

## Accessing the administrative server through a browser

This task describes how to access the administrative server through a browser from the master console (or from any computer that is on the same LAN as the SAN File System cluster).

Before accessing the administrative server through a browser from the master console, the following requirements must be met:

**Note:** You can access the SAN File System console from any computer that is on the same LAN as the SAN File System cluster.

- A SAN File System administrative user account must be set up for use in signing on to the SAN File System console.
- If the service representative is accessing the master console remotely:
  - Make sure that you have already initiated a VPN connection with the service representative.
  - The service representative must have established the VPN connection and used a remote display emulation package, such as VNC, to remotely view the desktop of the master console.

1. From the master console, open a Web browser and type the URL for the primary administrative server

`https://primary_administrative_server:7979/sfs`

where *primary\_administrative\_server* is the host name or IP address of the engine hosting the primary administrative server (as well as the master metadata server).

**Note:** If you enter a location in the Web browser for an administrative server other than the primary administrative server, the request is redirected to the primary administrative server. If the master administrative server is not available, the console for the secondary administrative server is displayed. However, some commands require that the primary administrative server be available; these commands will not complete successfully.

2. From the SAN File System console welcome page, enter a SAN File System administrative user name and password to sign on.

From the SAN File System console, you can manage and view information about engines, metadata servers, and clients.

## Adding a metadata server to the cluster

This topic describes how to add a metadata server to the cluster.

You must have Administrator or Operator privileges to perform this task.

The metadata server that you are adding must be in a state of "not added."

1. Click **Manage Servers and Clients**→**Servers** from the My Work frame.
2. Click **Add a Server to Cluster** from the drop-down box in the table header.

3. Click **Go**.
4. Enter the IP address of the metadata server in the IP Address field. If the cluster port is different than the default cluster port, you can append the port number to the IP address with a colon delimiter. For example, to specify an IP address of 9.67.101.202 and a port number of 1738, you can enter 9.67.101.202:1738.
5. Click **OK** to start the process.

## Changing the master metadata server

This topic describes how to change which server in the SAN File System cluster serves as the master.

You must have Operator or Administrator privileges to perform this task.

Before this task can be performed, the current master server must be down, but the cluster must be up. Additionally, the engine hosting the master server must be powered off. The subordinate server that will become the master must be in the “joining” or “forming” state only.

**Remember:** Before you shutdown the current master server, move the filesets that the current master server is serving to a subordinate server.

This task can be performed only at the administrative command-line interface. You may need to convert a subordinate server to the master server in the event of an irrecoverable loss of the original cluster master server. Such a loss might be due to hardware failures, software failures, or partitioned networks.

1. Go to the administrative command-line interface.
2. Make sure the cluster does not have a master metadata server: In the SAN File System console, if the master engine is running, you can check the **Servers** panel to make sure the master metadata server is down.

**Restriction:** If the master engine is not up, the SAN File System console will not work.

3. In certain failure scenarios it is possible, but unlikely, that the SAN File System console is not working, but the master metadata server is still up. You can try to log in to the operating system on the engine hosting the master metadata server and use the **lsserver** and a **stopserver** commands if necessary. In this scenario it is likely that the administrative command-line interface on the master engine also does not work, in which case, you should shut down the engine as described in a following step.
4. On the subordinate engine that you want to make the new master, a subordinate metadata server must be running. Log in to the operating system on this engine and use the **lsserver** command to see if a local subordinate metadata server is running. If not, using the **startserver** command on this engine starts the subordinate metadata server.
5. Next, make sure the engine that was hosting the master metadata server is shut down:
  - a. Use the **lsengine master\_metadata\_server** command to determine if the engine hosting the master metadata server is running.
  - b. Use the **stopengine master\_metadata\_server** command to shut it down.

**Tip:** You might need to use **-f** flag if the engine is hung. Use **lsengine** again to ensure it is down.

6. In the unlikely case that the failure is severe enough that these commands are not working, you must manually power off the engine that was hosting the master, and manually reboot the engine that you want to become master. Then return to step 4 on page 97.
7. Enter the **setmaster** command and the name of the server that you want to assign master status. You can test the success of this action by using the **lsserver** command. Once this step is successful, the SAN File System console will work.

**Attention:** If you receive note of an internal error after performing this operation, please contact your IBM Support personnel.

After issuing the **setmaster** command, you must manually edit the `tank.properties` file on the subordinate server that will become the master and set the `IsMaster` property to `true`.

## Changing metadata server hostname

This topic describes how to change the hostname of a metadata server in the `/etc/HOSTNAME` and `/etc/hosts` configuration files after the initial setup and configuration of the SAN File System.

1. Edit the `/etc/hosts` and `/etc/HOSTNAME` files with the new hostname.
2. From the command line on each metadata server, type:
  - a. `/bin/rm -fR /opt/was`
  - b. `/opt/bobcat_src/bobcat_install`
  - c. `reboot`
  - d. `/usr/tank/admin/bin/stopConsole`
  - e. `/usr/tank/admin/bin/installConsole`
  - f. `/usr/tank/admin/bin/startConsole`

## Changing metadata server IP address

This topic describes how to change the IP address for a metadata server.

This change requires the server to go offline and be rebooted.

1. Reassign statically-assigned filesets so the filesets are no longer served by the metadata server as necessary.
2. Drop the metadata server for which you want to change the IP address. You can use the **dropserver** command.
3. Reconfigure the engine that the metadata server resides on so that the engine uses the new IP address. You need to edit the appropriate Linux configuration files and issue the Linux **route** and **ipconfig** administrative commands.
4. Run the `setupsfs` utility to create a new instance of the metadata server on the engine and specify the new Server IP address.
5. Add the new metadata server to the cluster from the master metadata server using the **addserver** command or the **Add a server to Cluster** action from the **Servers** panel.

## Checking metadata

This topic describes how to check and repair metadata.

You must have Administrator privileges to check or repair the metadata.

1. Click **Maintain System**→**Check Metadata** from the My Work frame.

2. Fill in the appropriate fields.
3. Optionally, select the **Repair as necessary** check box to repair any corrupt metadata.
4. Click **OK** to start the process.
5. If you selected the **Repair as necessary** check box, click **OK** to confirm that you want to repair any corrupt metadata.

## Listing metadata servers

This topic describes how to list SAN File System metadata servers.

To display a list of SAN File System metadata servers, click **Manage Servers and Clients**→**Servers** from the My Work frame.

## Removing a metadata server from the cluster

This topic describes how to remove a metadata server from the cluster.

You must have Administrator or Operator privileges to perform this task.

Be aware that the filesets on this server are reassigned to a new server.

1. Click **Manage Servers and Clients**→**Servers** from the My Work frame.
2. Click **Remove from Cluster** from the drop-down box in the table header.
3. Click **Go**.

## Starting a metadata server

This topic describes how to start a metadata server.

You must have Administrator or Operator privileges to perform this task.

Starting a metadata server will attempt to put the server or servers selected in an available clustered state, if they are not already in such a state. You should not use the start action to recycle a server.

### Note:

1. If the master server is either “not running” or “unknown,” then starting subordinate servers, without also starting the master, is not allowed.
  2. The system will try to match the servers to the cluster’s current target state (online, offline, fully or partly quiescent).
1. Click **Manage Servers and Clients**→**Servers** from the My Work frame.
  2. Select one or more metadata servers to start.

**Note:** You can only start those metadata servers with a state of “not running” or “unknown.”

3. Click **Start** from the drop-down box in the table header.
4. Click **Go**.

## Starting the metadata server restart service

This topic describes how to enable the metadata server restart service.

You must have Operator or Administrator privileges to perform this task.

The metadata server restart service is enabled by default.

This service enables probes to periodically check the metadata server condition. By continually monitoring the server state, probes can automatically restart the metadata servers as appropriate.

**Remember:** If a metadata server has the restart service enabled, and the metadata server is later manually shut down, the restart service is disabled and you need to start it again as described in this task.

1. Click **Maintain System**→**Restart Service** from the My Work frame.
2. Select one or more metadata servers on which to enable the restart service.  
The **Service State** field indicates whether the service is currently enabled or disabled for a particular server.
3. Click **Enable Service** from the drop-down box in the table header.
4. Click **Go** to start the service on the selected server or servers.

## Stopping a metadata server

This topic describes how to stop a metadata server.

You must have Administrator or Operator privileges to perform this task.

This task allows you to gracefully stop the one or more selected metadata servers. The metadata servers and their associated states, roles, and filesets are displayed for review and confirmation before the stop is executed.

1. Click **Manage Servers and Clients**→**Servers** from the My Work frame.
2. Select one or more metadata servers to stop.

**Note:** Only active metadata servers can be stopped. Active metadata servers are those in one of the following states:

- offline
- fully quiescent
- partly quiescent
- online

3. Click **Stop** from the drop-down box in the table header.
4. Click **Go**.
5. Verify that you selected the metadata servers that you want to be stopped.
6. Click **OK** to confirm the stop action.

## Stopping the metadata check

This topic describes how to stop the metadata-check process.

You must have Administrator, Operator, or Backup privileges to stop the metadata-check process.

To stop the current metadata-check process, use the **stopmetadatacheck** command from the administrative CLI.

## Stopping the metadata server restart service

This topic describes how to disable the metadata server restart service.

You must have Operator or Administrator privileges to perform this task.

This service enables probes to periodically check the metadata server condition. By continually monitoring the server state, probes can automatically restart the metadata servers as appropriate. This task describes how to disable the service.

1. Click **Maintain System**→**Restart Service** from the My Work frame.
2. Select one or more metadata servers on which to disable the restart service.  
The **Service State** field indicates whether the service is currently enabled or disabled for a particular server.
3. Click **Disable Service** from the drop-down box in the table header.
4. Click **Go** to stop the service on the selected server or servers.

## Tuning the metadata server restart service

This topic describes how to set metadata server restart service tuning options such as retry limits, probe intervals, and liveness test timeouts.

This service enables probes to periodically check the metadata server condition. By continually monitoring the server state, probes can automatically restart the metadata servers as appropriate. This task describes how to set tuning options related to this service.

1. Click **Maintain System**→**Restart Service** from the My Work frame.
2. Select a metadata server for which you want to set restart service tuning options.

**Note:** The **Service State** field indicates whether the service is currently enabled or disabled for a particular server.

3. Click **Details** from the drop-down box in the table header.
4. Click **Go** to see the settings for the selected server.
5. Click the **Tuning** tab.
6. Change the current settings as desired and click **OK** to save the new settings.

## Viewing metadata server details

This topic describes how to view metadata server details such as state, server role, boot times, and software version. These details also include networking information such as port, IP address and protocol information.

1. Click **Manage Servers and Clients**→**Servers** from the My Work frame.
2. Select the server for which you want to view details.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Details** tab.

## Viewing metadata server statistics

This topic describes how to view metadata server statistics.

1. Click **Monitor System**→**Statistics** from the My Work frame.
2. Select the **Server** tab.
3. Click **OK** to view the statistics.
4. Click **Close** to close the Statistics panel.

## Viewing metadata server restart service statistics

This topic describes how to view metadata server restart service statistics, which includes test details and information about metadata server probes..



This service enables probes to periodically check the metadata server condition. By continually monitoring the server state, probes can automatically restart the metadata servers as appropriate. This task describes how to view statistics related to this service.

1. Click **Maintain System**→**Restart Service** from the My Work frame.
2. Select a metadata server for which you want to view restart service statistics.  
The **Service State** field indicates whether the service is currently enabled or disabled for a particular server.
3. Click **Statistics** from the drop-down box in the table header.
4. Click **Go** to view the statistics for the selected server.
5. Click the **Probe Overview** or **Test Details** tabs, if not already selected, depending on the type of information that you want to view.

## Viewing metadata server restart service tuning details

This topic describes how to view metadata server restart service tuning details such as retry limits, probe intervals, and liveness test timeouts.

This service enables probes to periodically check the metadata server condition. By continually monitoring the server state, probes can automatically restart the metadata servers as appropriate. This task describes how to view tuning information related to this service.

1. Click **Maintain System**→**Restart Service** from the My Work frame.
2. Select a metadata server for which you want to view restart service tuning details.  
The **Service State** field indicates if the service is currently enabled or disabled for a particular server.
3. Click **Details** from the drop-down box in the table header.
4. Click **Go** to view the tuning details for the selected server.
5. Click the **Details** tab.

## Removing down-level metadata server software

This topic describes how to remove down-level software from the metadata server.

1. Stop the engine.
2. Type **stopserver *server\_name***, and press Enter, where *server\_name* is the name or IP address of the engine to stop.
3. Type **rpm -e *package\_name***, and press Enter to remove the metadata server component software package, where *package\_name* is the file name of the metadata server package (for example, **rpm -e storagetank-server-RHLAS-1.0.0.rpm**).

---

## Managing policies

Policies and the rules that they contain are used to assign files to specific storage pools. A storage pool typically contains a set of volumes that provide a specific quality of service for a specific use, such as to store all files for a particular application or a specific business division.

Keep in mind that policies can reference file or path names in ASCII only.

## Activating a policy

This topic describes how to activate a policy.

You must have Administrator privileges to perform this task.

1. Click **Manage Filing→Policies** from the My Work frame.
2. Select the policy that you want to activate.
3. Click **Activate** from the drop-down box in the table header.
4. Click **Go**.
5. Click **OK** to confirm the change.

## Changing the rules in a policy

This topic describes how to modify the rules in a policy.

You must have Administrator privileges to perform this task.

Policy properties, including any associated rules, are stored in metadata. They are not stored in a file. Therefore, you cannot change the rules in the policy by updating the rules file that you used to initially create the policy.

Each policy has a length limit of 32 KB.

1. Click **Manage Filing→Policies** from the My Work frame.
2. Select the policy that you want to change.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Rules** tab.
6. Edit the existing rules or add new ones.
7. Click **Apply** to save the changes.

## Checking the placement of current files

This topic describes how to check how the rules in the policy are being applied to place files.

You must have Administrator privileges to perform this task.

1. Use the **lsvol** command to display a list of available volumes.  

```
sfscli> lsvol
```
2. Specify the **reportvolfiles** command for each of the available volumes, to display a list of its files (and their attributes).  

```
sfscli> reportvolfiles volume-name
```

## Copying a policy

This topic describes how to copy an existing policy.

You must have Administrator privileges to perform this task.

SAN File System provides a wizard to step you through the process of creating a copy of an existing policy.

To start the create-policy wizard, click **Manage Filing→Create a Policy** from the My Work frame. Select the Clone Policy radio button, and then select the policy that you want to copy from the Existing Policy drop-down list.

## Creating a policy

This topic describes how to create a policy.

You must have Administrator privileges to perform this task.

SAN File System console provides a wizard to step you through the process of creating a policy.

Policy properties, including any associated rules, are stored in metadata. They are not stored in a file.

1. Start the Create-policy wizard by clicking **Manage Filing→Create a Policy** in the My Work frame.
2. Click **Next**.
3. In the Create a Policy panel under **High-Level settings**, fill in the Name and Description for the policy. Then click **Next**.
4. Under **Add Rules**, fill in the **Rules Description** field with a description of the rule.
5. Select a storage pool from the **Storage Pool Assignment** drop-down list.
6. Choose the rule specifics.
7. Continue creating all rules for this policy, and click **Next** when finished.
8. In the Edit Rules for Policy panel, verify the rules.
9. Click **Manage Filing→Policies**
10. Select the policy you just created, and click **Activate** from the drop-down list.
11. Click **Go** to activate the policy and verify the activation.

## Deleting a policy

This topic describes how to delete a policy.

You must have Administrator privileges to perform this task.

1. Click **Manage Filing→Policies** from the My Work frame.
2. Select the policy that you want to delete.
3. Click **Delete** from the drop-down box in the table header.
4. Click **Go**.
5. Click **OK** to confirm the deletion.

## Listing policies

This topic describes how to display a list of all policies.

To display a list of all policies, click **Manage Filing→Policies** from the My Work frame.

## Changing the name of a policy

This topic describes how to change the policy settings.

You must have Administrator privileges to perform this task.

You cannot change the name of a policy. Instead, clone the existing policy, and then delete the original one.

## Viewing policy details

This topic describes how to view details for a specific policy.

1. Click **Manage Filing→Policies** from the My Work frame.
2. Select a policy.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Details** tab.

## Viewing policy rules

This topic describes how to view the rules in a specific policy.

1. Click **Manage Filing→Policies** from the My Work frame.
2. Select a policy.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Rules** tab.

## Viewing policy statistics

This topic describes how to view policy statistics.

1. Click **Manage Filing→Policy Statistics** from the My Work frame.
2. Click the **Per Rule** tab to view statistics about the number of times specific rules have been evaluated and applied, or click the **Per Policy** tab to view statistics about files placed in storage pools.

---

## Managing processes

You can configure and view SAN File System processes (also known as “threads.”)

## Listing processes

This topic describes how to display a list of long-running processes.

To display a list of all long-running processes, click **Monitor System→Processes** from the My Work frame.

## Viewing process details

This topic describes how to display the details for a long-running process.

1. Click **Monitor System→Processes** from the My Work frame.
2. Select a process.
3. Click **General Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Details** tab.

## Viewing process limits

This topic describes how to display the limits for a specific long-running process.

1. Click **Monitor System→Processes** from the My Work frame.
2. Select a process.
3. Click **General Properties** from the drop-down box in the table header.
4. Click **Go**.

5. Click the **Process Limits** tab.

---

## Managing storage pools

A storage pool is a collection of SAN File System volumes that can be used to store either metadata or file data. A storage pool typically contains a set of volumes that provide a desired quality of service for a specific use, such as to store all files for a particular application or a specific business division. An administrator must assign one or more volumes to a storage pool before it can be used.

### Changing storage pool settings

This topic describes how to modify the general settings for a specific storage pool.

You must have Administrator privileges to perform this task.

1. Click **Manage Storage→Pools** from the My Work frame.
2. Select a storage pool.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **General Settings** tab.
6. Edit the appropriate fields.
7. Click **OK** to save the new settings.

### Creating a storage pool

This topic describes how to create a storage pool.

You must have Administrator privileges to perform this task.

SAN File System console provides a wizard to step you through the process of creating a storage pool. This wizard creates volumes and assigns the volumes to the storage pool.

To create a storage pool from the administrative command-line interface, you must first create the volumes using the **mkvol** command, and then create the storage pool using the **mkpool** command.

1.
  - To start the wizard to create a storage pool, click **Manage Storage→Create a Pool** from the My Work frame. View the list of steps to create a storage pool and click **Next**.
2. Under **Pool Settings** in the Create a Storage Pool panel, fill in the **Name** field (for example, My\_New\_Pool) and the **Description** field (for example, A pool I created). Optionally, select **Logical Partition Size**, **Allocation Size** and **Usage Threshold**. Click **Next**. Clicking **Toggle Help Assistant** after entering values in the fields clears the fields.
3. Under **Select Client**, select a client and a fetch method to gather the available LUNs information for the next step, adding volumes to the storage pool. Click **Next**.
4. Under **Add Volumes**, select an LUN (for example, VPD83NAA6=60050768018200281800000000000036) from the table. Click **Next**.
5. Under **Volume Settings**, fill in the **Volume Name Prefix** field (for example, My\_New\_Pool), and click **Next**.
6. Verify your settings, and then click **Finish**.

7. Click **Manage Storage→Pools** from the My Work frame to verify that the storage pool was created.

## Deleting a storage pool

This topic describes how to delete a storage pool.

You must have Administrator privileges to perform this task.

1. Click **Manage Storage→Pools** from the My Work frame.
2. Select the storage pool that you want to delete.
3. Click **Delete** from the drop-down box in the table header.
4. Click **Go**.
5. Click **OK** to confirm the deletion.

## Disabling the default storage pool

This topic describes how to disable the default storage pool.

You must have Administrator privileges to perform this task.

1. Click **Manage Storage→Pools** from the My Work frame.
2. Select the storage pool that you want to be the default storage pool.
3. Click **General Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click **Disable** to disable the use of the default storage pool.
6. Click **OK** to confirm the change.

## Listing storage pools

This topic describes how to display a list of all storage pools.

To display a list of all storage pools, click **Manage Storage→Pools** from the My Work frame.

## Listing volumes in a storage pool

This topic describes how to display a list of the volumes in a storage pool.

1. Click **Manage Storage→Pools** from the My Work frame.
2. Select a storage pool.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Current Volumes** tab.

## Setting the default storage pool

This topic describes how to set the default storage pool.

You must have Administrator privileges to perform this task.

1. Click **Manage Storage→Pools** from the My Work frame.
2. Select the storage pool that you want to be the default storage pool.
3. Click **General Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click **Enable** and select the storage pool to be enabled as the default storage pool from the drop-down box.

6. Click **OK** to confirm the change.

## Viewing storage pool details

This topic describes how to view details for a specific storage pool.

1. Click **Manage Storage**→**Pools** from the My Work frame.
2. Select a storage pool.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Details** tab.

## Viewing storage pool statistics

This topic describes how to view storage pool statistics.

1. Click **Monitor System**→**Statistics** from the My Work frame.
2. Select the **Storage Pools** tab.
3. Click **OK** to view the statistics.
4. Click **Close** to close the Statistics panel.

---

## Managing users

SAN File System provides different levels of user access that you can assign to administrative tasks in your environment. User roles must be configured at an LDAP server.

## Granting SAN File System user access and roles

This topic describes how to grant access and roles for administrative users of SAN File System.

You must have access to your LDAP server to perform this task.

SAN File System authenticates users by taking their names (with role mappings) and associated passwords from an LDAP server. Therefore, you cannot grant anyone administrative access to SAN File System without first defining them on your LDAP server.

User name, user-to-role mapping and passwords must be entered at the LDAP server, employing the schemas of the LDAP server in your environment. Refer to your LDAP server documentation for more information.

**Note:** User roles and associated level of read, write, and execute access are defined as a function of your business requirements during the planning phase.

## Listing user roles

This topic describes how to list user roles currently defined within SAN File System.

To view a listing of user roles currently defined within SAN File System, click **Administer Access**→**Roles** from the My Work frame.

## Listing users

This topic describes how to list users with administrative access to SAN File System.



To view a listing of users with administrative access to SAN File System, along with their associated user roles, click **Administer Access→Users** from the My Work frame.

## Timing out all user authorizations

This topic describes how to timeout administrative user authorizations, thereby requiring all users to re-authenticate with the LDAP server.

You must have Administrator privileges to perform this task.

1. Click **Administer Access→Users** from the My Work frame.
2. Click **Timeout All Authorizations** from the drop-down box in the table header.
3. Click **Go** to confirm the timeout action.

---

## Managing volumes and LUNs

A LUN becomes a SAN File System volume when an administrator adds it to a storage pool. It is automatically assigned a system-generated label that identifies it as a SAN File System volume. An administrator must also give the volume a name that is unique within all the volumes used by a SAN File System cluster.

You can perform the following tasks for volumes and LUNs:

### Activating a volume

This topic describes how to activate a volume.

You must have Administrator privileges to perform this task.

1. Click **Manage Storage→Volumes** from the My Work frame.
2. Select a volume.
3. Click **Activate** from the drop-down box in the table header.
4. Click **Go**.

### Adding LUNs to SAN File System

This topic describes how to add LUNs to SAN File System.

This task describes how to add LUNs without having to stop the cluster, stop and restart each metadata server, and start the cluster.

1. Assign the new LUNs to the appropriate zones, according to LUN restrictions and limitations and the considerations for setting up zones.
2. On each metadata server, to force the QLogic driver to rescan the fabric, enter the following commands at the system prompt:  

```
echo scsi-qlasscan >/proc/scsi/qla2300/2
echo scsi-qlasscan >/proc/scsi/qla2300/3
```
3. View the target ID and the LUN for all LUNs that were added. Enter the following command at the system prompt:  

```
more /proc/scsi/qla2300/2
```
4. Record the ID:LUN combination for any LUN that is not yet registered with the operating system.
5. For each controller number (2 and 3 are the controller numbers for the QLogic 2342 ports) and ID:LUN combination, enter the following command at the system prompt:

```
echo "scsi add-single-device controller 0 Id Lun ">/proc/scsi/scsi
```

6. To force the Subsystem Device Driver (SDD) to rescan and map the new devices, enter the following command at the system prompt:  
`/usr/sbin/cfgvpath`
7. To map the block devices to character devices, enter the following command at the system prompt:  
`/usr/tank/server/bin/device_init.sh`
8. To verify that the LUN was added before a metadata server is up, you can enter the **cat proc/partitions** command. After a metadata server is up, you can enter the **ls lun** command to view the list of LUNs available to SAN File System.
9. For each client for AIX, type **stfsdisk** and press Enter to force the client to discover the new volume. For each client for Windows, click Disk Rescan in disk management to discover the new storage.

## Adding volumes to a storage pool

This topic describes how to add volumes to a storage pool.

You must have Administrator privileges to perform this task.

SAN File System provides a wizard to step you through the process of adding volumes to a storage pool.

To start the wizard to add a volume, click **Manage Storage**→**Add Volumes** from the My Work frame.

## Changing volume settings

This topic describes how to modify the general settings for a specific volume.

You must have Administrator privileges to perform this task.

1. Click **Manage Storage**→**Volumes** from the My Work frame.
2. Select a volume.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **General Settings** tab.
6. Edit the appropriate fields.
7. Click **OK** to save the new settings.

## Increasing volume size

This topic describes how to increase the size of a specific volume when the corresponding LUN size has increased.

You must have Administrator privileges to perform this task.

1. Click **Manage Storage**→**Volumes** from the My Work frame.
2. Select a volume.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Size** tab.
6. Determine the current LUN size by selecting a client that can see the LUN from the drop-down box and clicking **Refresh**.

7. Check the LUN size versus the volume size in the table and click **Expand Volume** to increase the volume size if the LUN size has increased.
8. Click **OK** to save the new settings.

## Listing available LUNs

This topic describes how to display a list of the available LUNs.

You can choose to display a list of data LUNs or a list of metadata LUNs:

- To display a list of data LUNs in the SAN that are available to add to user storage pools, click **Manage Storage→Available Data LUNs** from the My Work frame.
- To display a list of metadata LUNs in the SAN that are available to add to the system storage pool, click **Manage Storage→Available Metadata LUNs** from the My Work frame.

## Listing files on a volume

This topic describes how to display a list of files on a volume.

You must have Administrator, Operator, or Backup privileges to perform this task.

To display a list of files on a specific volume, run the `reportvolfiles` command from the administrative CLI.

## Listing LUNs

This topic describes how to display a list of all data LUNs or all metadata LUNs.

You can choose to display a list of all data LUNs or a list of all metadata LUNs:

- To display a list of all data LUNs in the SAN, click **Manage Storage→Data LUNs** from the My Work frame.
- To display a list of all metadata LUNs in the SAN, click **Manage Storage→Metadata LUNs** from the My Work frame.

## Listing volumes

This topic describes how to display a list of all volumes.

To display a list of all volumes, click **Manage Storage→Volumes** from the My Work frame.

## Managing free and allocated storage space

This topic describes how to manage free and allocated space.

You can use the storage pool usage threshold and the fileset quota and usage threshold to manage the storage space in SAN File System.

The storage pool usage threshold is a percentage of the storage pool's estimated storage capacity. When the amount of used space reaches or exceeds the specified threshold, the metadata server generates an alert.

The fileset quota is the size limit, in megabytes for the fileset. When the size of the fileset reaches or exceeds the specified limit, the metadata server generates an alert. You can specify either a hard or soft quota. A hard quota denies requests for additional space, and a soft quota grants requests space allocation.

**Note:** The quota must be set to a whole number.

The fileset usage threshold is a percentage of the fileset's quota. When the amount of used space reaches or exceeds the specified threshold, the metadata server generates an alert.

1. Perform the following steps to configure the storage pool threshold:
  - a. Click **Manage Storage→Pools** from the My Work frame.
  - b. Select a storage pool.
  - c. Click **Properties** from the drop-down box in the table header.
  - d. Click **Go**.
  - e. Click the **General Settings** tab.
  - f. Verify that the **Enable usage alerts** check box is selected.
  - g. Type a percentage in the **Usage Threshold** field.
  - h. Click **OK** to save the new settings.
2. Perform the following steps to configure the fileset quota:
  - a. Click **Manage Filing→Filesets** from the My Work frame.
  - b. Select a fileset.
  - c. Click **Properties** from the drop-down box in the table header.
  - d. Click **Go**.
  - e. Click on the **General Settings** tab.
  - f. Specify the type of quote by selecting either a hard or soft quota in the **Quota Type** drop-down list.
  - g. Type a size limit in the **Quota Size** field.

**Note:** If a hard quota already exists, you must not set the new size to a value less than the current quota size for this fileset.

- h. Verify that the **Enable usage alerts** check box is selected.
- i. Type a percentage in the **Usage Threshold** field.
- j. Click **OK** to save the new settings.

## Removing volumes from a storage pool

This topic describes how to remove volumes from a storage pool.

You must have Administrator privileges to perform this task.

1. Click **Manage Storage→Volumes** from the My Work frame.
2. Select one or more volumes that you want to remove.
3. Click **Remove** from the drop-down box in the table header.
4. Click **Go**.
5. Select a client that can see the volume and the removal method.
6. Click **OK**.

## Suspending a volume

This topic describes how to suspend a volume.

You must have Administrator privileges to perform this task.

1. Click **Manage Storage→Volumes** from the My Work frame.
2. Select a volume.
3. Click **Suspend** from the drop-down box in the table header.

4. Click **Go**.

## Viewing available LUN details

This topic describes how to view details for a specific available LUN.

1. Click **Manage Storage**→**Available LUNs** from the My Work frame.
2. Select an available LUN.
3. Click **Details** from the drop-down box in the table header.
4. Click **Go**.

## Viewing LUN details

This topic describes how to view details for a specific LUN.

1. Click **Manage Storage**→**LUNs** from the My Work frame.
2. Select a LUN.
3. Click **Details** from the drop-down box in the table header.
4. Click **Go**.

## Viewing LUN statistics

This topic describes how to view LUN statistics.

1. Click **Monitor System**→**Statistics** from the My Work frame.
2. Select the **LUNs** tab.
3. Click **OK** to view the statistics.
4. Click **Close** to close the Statistics panel.

## Viewing volume details

This topic describes how to display the details for a volume.

1. Click **Manage Storage**→**Volumes** from the My Work frame.
2. Select a volume.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **Details** tab.

## Viewing volume settings

This topic describes how to display the general settings for a volume.

1. Click **Manage Storage**→**Volumes** from the My Work frame.
2. Select a volume.
3. Click **Properties** from the drop-down box in the table header.
4. Click **Go**.
5. Click the **General Settings** tab.

## Viewing volume statistics

This topic describes how to view volume statistics.

1. Click **Monitor System**→**Statistics** from the My Work frame.
2. Click the **Volume** tab.
3. Click **OK** to view the statistics.
4. Click **Close** to close the Statistics panel.



---

## Appendix A. Accessibility

This topic provides information about the accessibility features of SAN File System and its accompanying documentation.

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully.

### Features

These are the major accessibility features in SAN File System:

- You can use screen-reader software and a digital speech synthesizer to hear what is displayed on the screen.

**Note:** The SAN File System Information Center and its related publications are accessibility-enabled for the IBM Home Page Reader.

- You can operate all features using the keyboard instead of the mouse.

### Navigating by keyboard

You can use keys or key combinations to perform operations and initiate many menu actions that can also be done with a mouse. You can navigate the SAN File System console and help system from the keyboard by using the following key combinations:

- To traverse to the next link, button or topic, press Tab inside a frame (page).
- To expand or collapse a tree node, press Right Arrow or Left Arrow, respectively.
- To move to the next topic node, press Down Arrow or Tab.
- To move to the previous topic node, press Up Arrow or Shift+Tab.
- To scroll all the way up or down, press Home or End, respectively.
- To go back, press Alt+Left Arrow
- To go forward, press Alt+Right Arrow.
- To go to the next frame, press Ctrl+Tab. There are quite a number of frames in the help system.
- To move to the previous frame, press Shift+Ctrl+Tab.
- To print the current page or active frame, press Ctrl+P.





---

## Appendix B. Commands

SAN File System has two sets of commands: administrative and client commands.

### Administrative commands

The administrative commands run on the storage engines that host the metadata server. Most commands must be run from the master metadata server. There are a few commands that must be run from subordinate metadata server for specific situations.

You run a majority of the administrative commands from the sfscli session to manage SAN File System. There are a few commands that must be run from the operating-system shell prompt.

To use the administrative commands, you must log in directly to the engine, or from another workstation through SSH, using the local operating system authentication mechanism. You must then log in to the administrative server on the engine using the same administrative user ID and password that you would use to log into the SAN File System console. You can specify the password in one of two ways:

- Set the password in the sclif.properties file, located in your home directory on the engine (for example, joe/sclif.properties), to your valid LDAP password using the tankpasswd utility.
- Set the SFS\_CLI\_PASSWDFILE environment variable to the location of the password file.

When you run administrative commands that take a long time to complete, in a system with active applications, those applications that are sensitive to the response time of the system might experience timeout errors. An example of a possible long running command is **quiescecluster**.

**Tip:** The administrative commands are case sensitive. If you enter a command in uppercase, you receive an error.

### Client commands

The client commands runs on any client machine on which the client file-system driver has been installed. It provides a set of commands that you can use to manage your clients.

To use the client commands, you must log in directly to the client machine or from another workstation using SSH. You log in using the user ID and password for the client machine. You must have administrative (Windows) or root (UNIX-based) privileges to use the client commands.

---

## Administrative CLI overview

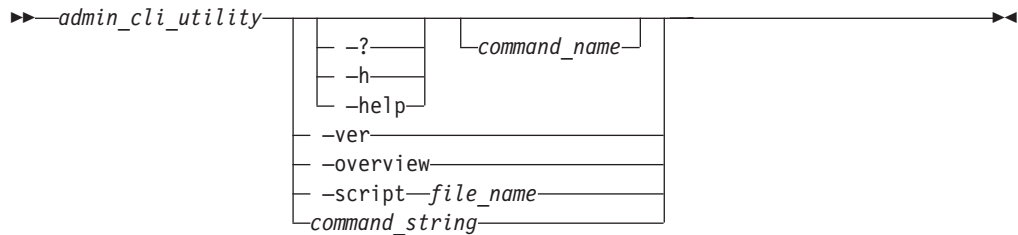
The SAN File System administrative command-line interface (CLI) provides a method for managing SAN File System.

## Administrative command-line interface

The administrative command-line interface (CLI) utility allows you to run administrative commands in interactive mode. You can also run a single command or run a set of commands from a script without starting an interactive administrative CLI session.

If you run the administrative CLI utility with any of the valid parameters, an interactive session is not started. If you run the command with no parameters, the administrative CLI starts in a session. When you are in an interactive administrative CLI session, the *admin\_cli\_utility>* prompt is displayed.

This is the syntax for using the administrative CLI command:



### Parameters

**`-? | -h | -help`** *command\_name*

Displays help for the specified command (for example, **`-h catlog`** displays help for the **catlog** command). If a command name is not specified, this parameter displays a list of available commands in the administrative CLI.

**`-ver`**

Displays the current version and licensing information for this product.

**`-overview`**

Displays the overview information about the administrative CLI, including command modes, standard format, help and listing parameters, and syntax diagram conventions.

**`-script`** *file\_name*

Runs the set of command strings in the specified file outside of an interactive administrative CLI session. If you specify this parameter, you must specify a file name.

The format options specified using the **setoutput** command apply to all commands in the script.

Output from successful commands routes to the standard output stream (stdout). Output from unsuccessful commands route to the standard error stream (stderr). If an error occurs while one of the commands in the script is running, the script will exit at the point of failure and return to the system prompt.

*command\_string*

Runs the specified command string outside of an administrative CLI session.

## Command modes

You can work with the administrative CLI in one of three modes: single-shot, interactive, and script.

## Single-shot mode

If you want to run only a single command, specify the administrative CLI utility and the command that you want to run from the shell prompt, for example:

```
shell> admin_cli_utility lspool
DEFAULT,Default,10000,2500,25,80,10,64,Default Storage Pool
admin_cli_utility> exit
shell>
```

## Interactive mode

If you want to run several commands, start an administrative CLI session using the administrative CLI utility with no parameters, and then enter each command at the *admin\_cli\_utility>* prompt, for example:

```
shell> admin_cli_utility
admin_cli_utility> lspool -l -type default
DEFAULT,Default,10000,2500,25,80,10,64,Default Storage Pool
admin_cli_utility> exit
shell>
```

## Script mode

If you want to run a set of commands that you defined in a file, use the administrative CLI utility with the **-script** parameter, for example:

```
shell> admin_cli_utility -script ~/bin/listpools.bat
```

You can add comments to the script file by placing a pound sign (#) in the first column, for example:

```
This script file lists the default storage pool.
lspool -l -type default
```

**Note:** Output from successful commands routes to the standard output stream (stdout). Output from unsuccessful commands route to the standard error stream (stderr). If an error occurs while one of the commands in the script is running, the script will exit at the point of failure and return to the system prompt.

## Naming guidelines

This topic provides guidelines to help you define objects and descriptions used in the administrative CLI.

### Objects

Use the following guidelines when specifying names for objects:

- You can use alphanumeric characters, dashes (-), underscores (\_), and periods (.) in the object names; however, object names cannot start with a dash or underscore character and must contain at least one alphanumeric character.
- The object name must not contain blank spaces.
- Most object names can contain up to 256 characters; the exception is cluster and metadata server names, which can contain up to 32 characters.
- Object names are case-sensitive.

### Descriptions

Use the following guidelines when specifying descriptions:

- A description can contain up to 256 characters and cannot start with a blank space.
- For administrative commands, if a description contains spaces, enclose the description in single quotation marks (') or double quotation marks (").
- For administrative commands, if a description contains quotation marks, enclose the description in opposite quotation marks (for example, 'This is a pool named \"Foo\"').
- For administrative commands, if a description that is enclosed in quotation marks also contains single quotation marks ('), double quotation marks ("), or asterisks (\*) within the string, precede the character with a backslash (for example, "\"\*\"\*This is a test\"\*\"\*").

## Ports

You can specify a value from 1 024 to 65 535 for port numbers.

## Host and domain names

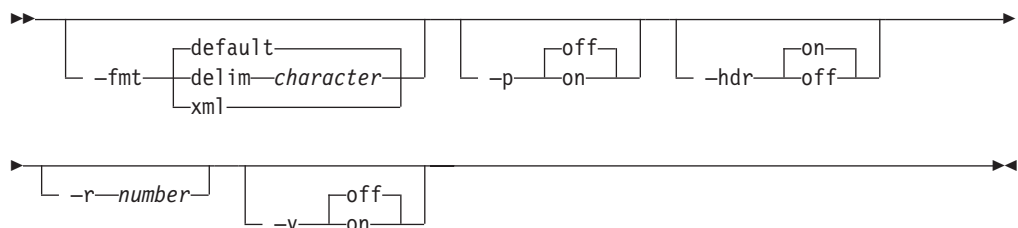
Machine host names must not include the underscore character (\_). Internet standards dictate that domain names conform to the host name requirements described in Internet Official Protocol Standards RFC 952 and RFC 1123.

Domain names must contain only letters (upper or lower case) and digits. Domain names can also contain dash characters ( - ) as long as the dashes are not on the ends of the name.

## Standard format parameters

The standard format parameters set the output format of listing commands (commands that start with ls\*) in the administrative CLI. These parameters can be used either in a listing command syntax or in the **setoutput** command syntax. You can specify these parameters in addition to the parameters available for a specific listing command.

The format settings remain in effect for the duration of the administrative CLI session or until you reset the parameters either by specifying these parameters in a listing command or using the **setoutput** command.



## Parameters

### -fmt

Specifies the format of the output. You can specify one of the following values:

#### default

Specifies to display output in a tabular format using spaces as the delimiter between the columns. This is the default value. For example:

| Name    | Type    | Size (GB) | Used (GB) | Used (%) | Alert (%) |
|---------|---------|-----------|-----------|----------|-----------|
| DEFAULT | Default | 10000     | 2500      | 25       | 80        |

| Volumes | Partition Size (MB) | Description          |
|---------|---------------------|----------------------|
| 10      | 64                  | Default Storage Pool |

**delim** *character*

Specifies to display output in a tabular format using the specified character to separate the columns. If you use a shell metacharacter (for example, \* or \t) as the delimiting character, enclose the character in single quotation marks (') or double quotation marks ("). A blank space is not a valid character. For example:

```
DEFAULT,Default,10000,2500,25,80,10,64,Default Storage Pool
```

**xml** Specifies to display output using XML format, for example:

```
<IRETURNVALUE>
<INSTANCE CLASSNAME="STC_StoragePool">
<PROPERTY NAME="Name" TYPE="string"><VALUE>DEFAULT_POOL</VALUE>
</PROPERTY>
<PROPERTY NAME="PoolType" TYPE="uint32"><VALUE>1</VALUE>
</PROPERTY>
<PROPERTY NAME="PartitionSize" TYPE="uint64"><VALUE>16</VALUE>
</PROPERTY>
<PROPERTY NAME="AlertPercentage" TYPE="uint16"><VALUE>80</VALUE>
</PROPERTY>
<PROPERTY NAME="Size" TYPE="uint64"><VALUE>0</VALUE></PROPERTY>
<PROPERTY NAME="SizeAllocated" TYPE="uint64"><VALUE>0</VALUE>
</PROPERTY>
<PROPERTY NAME="SizeAllocatedPercentage" TYPE="uint16"><VALUE>0
</VALUE></PROPERTY>
<PROPERTY NAME="NumberOfVolumes" TYPE="uint32"><VALUE>0</VALUE>
</PROPERTY>
<PROPERTY NAME="Description" TYPE="string"><VALUE>Default storage pool
</VALUE></PROPERTY>
</INSTANCE>
</IRETURNVALUE>
```

**-p** Specifies whether to display one page of text at a time or all text at once.

**off** Displays all text at one time. This is the default value when the administrative CLI is run in single-shot mode.

**on** Displays one page of text at time. Pressing any key displays the next page. This is the default value when the administrative CLI is run in interactive mode.

**-hdr**

Specifies whether to display the table header.

**on** Displays the table header. This is the default value.

**off** Does not display the table header.

**-r** *number*

Specifies the number of rows per page to display when the -p parameter is on. The default is 24 rows. You can specify a value from 1 to 100.

**-v** Specifies whether to enable verbose mode.

**off** Disables verbose mode. This is the default value.

**on** Enables verbose mode.

## Standard help parameters

The standard help parameters display user assistance for any command in the administrative CLI. Note that when you use a help parameter, all other parameters are ignored.

These are the help parameters:



### Parameters

**-? | -h | -help**

Displays the detailed description of a specific command.

## Standard listing parameters

The standard listing parameters specify whether to display the default, long, or short output for listing commands (commands that start with ls\*) in the administrative CLI. You can specify these parameters in addition to the parameters available for a specific listing commands.

If you do not specify a listing parameter, the default listing displays all objects and the most vital column information, for example:

Name	User Role	Authorization
JohnDoe	Admin	Current
MaryBlack	Backup	Not Current
JimSmith	Operator	Current
TomJones	Monitor	Not Current

The format set using these parameters remains in effect for the duration of the command.

These are the listing parameters:



### Parameters

**-s** Displays the list of objects with minimal information, for example,

```
Name
=====
JohnDoe
MaryBlack
JimSmith
TomJones
```

**-l** Displays the list of all objects with detailed information, for example:

Name	User Role	Authorization	Authorization Timeout (Secs)
JohnDoe	Admin	Current	300



MaryBlack	Backup	Not Current	0
JimSmith	Operator	Current	465
TomJones	Monitor	Not Current	0

## Syntax diagram conventions

### Syntax diagram conventions

To read syntax diagrams, follow the path of the line. Read from left to right, and top to bottom.

- The ► symbol indicates the beginning of the syntax diagram.
- The → symbol at the end of a line indicates that the syntax diagram continues on the next line.
- The ► symbol at the beginning of a line indicates that the syntax diagram continues from the previous line.

The →◄ symbol indicates the end of the syntax diagram.

Syntax diagrams use *position* to indicate required, optional, and default values for keywords, variables, and operands:

- On the line (required element)
- Above the line (default element)
- Below the line (optional element)

### Abbreviations

The administrative CLI does not currently support aliases or abbreviations; however, you can use aliases that you set up in the shell environment.

### Dash

A dash (-) indicates that you want to supply parameters from an input stream (stdin) rather than entering parameters. In the following example, the command line gets input from the file /work/myfile:

```
cmd - >> work/myfile
```

In the following example, the command line gets input from the keyboard:

```
cmd - >>
parameter_1
parameter_2
```

### Defaults

Default values are above the main line. If the default is a keyword, it appears only above the main line. You can specify this item or allow it to default. In the following example, the keyword A is the default. You can override it by choosing B or C. You can also specify the default value explicitly.



If an operand has a default value, the operand appears both above and below the main line. A value below the main line indicated that if you specify the operand, you must also specify either the default value or another value shown. If you do

not specify an operand, the default value above the main line is used. In the following example, the operand A=\* is the default. You can override it by choosing A=C. You can also specify the default value explicitly.



### Optional items

When one or more items are below the main line, all of the items are optional. In the following example, you can choose A, B, C, or nothing at all.



### Repeatable items

An arrow returning to the left means you can repeat the item, for example:



A character or space within the arrow means you must separate repeated items with that character or space, for example:



A stack of items followed by an arrow returning to the left means that you can select more than one item or, in some cases, repeat a single item. In the following example, you can choose any combination of A, B, or C.



### Required items

When a keyword, variable, or operand appears on the main line, you must specify that item. In the following example, you must choose A and B.

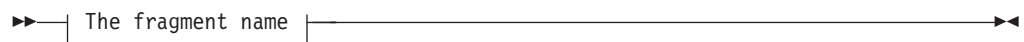


When two or more items are in a stack and one of them is on the main line, you must specify one item. In the following example, you must choose A, B, or C.



### Syntax fragments

Commands that contain lengthy groups or a section that is used more than once in a command are shown as separate fragments following the main diagram. The fragment name appears between vertical bars in the diagram. The expanded fragment also appears between vertical bars after the heading with the same fragment name.



### The



### Variables

Italicized, lowercase elements denote variables. In the following example, you must specify a variable name when you enter the keyword command:



## Common commands

These commands are common across other administrative command-line interfaces (CLIs).

### exit

Ends an administrative command-line interface (CLI) session.



### Parameters

#### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

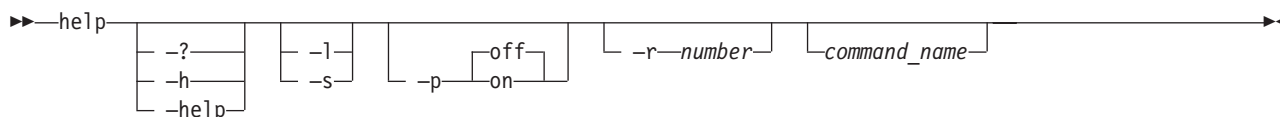
## Example

**Ends an administrative command-line interface (CLI) session** The following example ends the CLI session:

```
admin_cli_utility> exit
shell>
```

## help

Displays a list of commands available in an administrative command-line interface (CLI) and optionally displays the syntax or brief description of each command.



## Parameters

### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

**-l** Displays a list of available commands with the syntax diagrams for each. If you specify a command name with this parameter, this command displays the syntax for only the specified command.

**-s** Displays a list of available commands with a brief description of each. If you specify a command name with this parameter, this command displays a brief description for only the specified command.

**-p** Specifies whether to display one page of text at a time or all text at once.

**off** Displays all text at one time. This is the default value.

**on** Displays one page of text at time. Pressing any key displays the next page.

### -r number

Specifies the number of rows per page to display when the **-p** parameter is on. The default is 24 rows. You can specify a value from 1 to 100.

### command\_name

Displays help information for the specified command, including the syntax diagram, parameter descriptions, return codes and errors, descriptions, examples, and miscellaneous remarks.

## Description

If you specify this command with no parameters, this command displays only a list of available commands.

## Example

**Display a description of a command** The following example displays the description of the **mkimage** command:

```
admin_cli_utility> help -s lspool
lspool Displays a list of existing storage pools and their attributes.
```

quit

Ends an administrative command-line interface (CLI) session.



Parameters

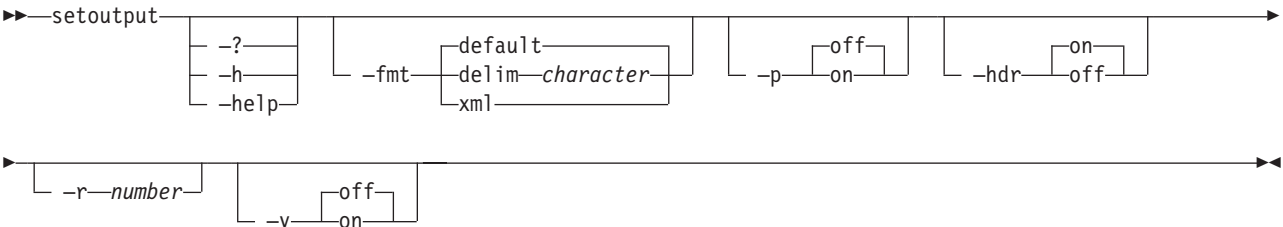
**-? | -h | -help**  
Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

Example

**Ends an administrative command-line interface (CLI) session** The following example ends the CLI session:  
`admin_cli_utility> quit`  
`shell>`

setoutput

Sets the output format for the administrative command-line interface (CLI).



Parameters

**-? | -h | -help**  
Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

**-fmt**  
Specifies the format of the output. You can specify one of the following values:

**default**  
Specifies to display output in a tabular format using spaces as the delimiter between the columns. This is the default value. For example:

Name	Type	Size (GB)	Used (GB)	Used (%)	Alert (%)
DEFAULT	Default	10000	2500	25	80
Volumes	Partition Size (MB)	Description			
10	64	Default Storage Pool			

**delim character**  
Specifies to display output in a tabular format using the specified character to separate the columns. If you use a shell metacharacter (for

example, \* or \t) as the delimiting character, enclose the character in single quotation mark (') or double quotation mark ("). A blank space is not a valid character. For example:

```
DEFAULT,Default,10000,2500,25,80,10,64,Default Storage Pool
```

**xml** Specifies to display output using XML format, for example:

```
<IRETURNVALUE>
<INSTANCE CLASSNAME="STC_StoragePool">
<PROPERTY NAME="Name" TYPE="string"><VALUE>DEFAULT_POOL</VALUE>
</PROPERTY>
<PROPERTY NAME="PoolType" TYPE="uint32"><VALUE>1</VALUE>
</PROPERTY>
<PROPERTY NAME="PartitionSize" TYPE="uint64"><VALUE>16</VALUE>
</PROPERTY>
<PROPERTY NAME="AlertPercentage" TYPE="uint16"><VALUE>80</VALUE>
</PROPERTY>
<PROPERTY NAME="Size" TYPE="uint64"><VALUE>0</VALUE></PROPERTY>
<PROPERTY NAME="SizeAllocated" TYPE="uint64"><VALUE>0</VALUE>
</PROPERTY>
<PROPERTY NAME="SizeAllocatedPercentage" TYPE="uint16"><VALUE>0
</VALUE></PROPERTY>
<PROPERTY NAME="NumberOfVolumes" TYPE="uint32"><VALUE>0</VALUE>
</PROPERTY>
<PROPERTY NAME="Description" TYPE="string"><VALUE>Default storage pool
</VALUE></PROPERTY>
</INSTANCE>
</IRETURNVALUE>
```

**-p** Specifies whether to display one page of text at a time or all text at once.

**off** Displays all text at one time. This is the default value when the **cli** command is run in single-shot mode.

**on** Displays one page of text at time. Pressing any key displays the next page. This is the default value when the **cli** command is run in interactive mode.

**-hdr**

Specifies whether to display the table header.

**on** Displays the table header. This is the default value.

**off** Does not display the table header.

**-r number**

Specifies the number of rows per page to display when the **-p** parameter is on. The default is 24 rows. You can specify a value from 1 to 100.

**-v** Specifies whether to enable verbose mode.

**off** Disables verbose mode. This is the default value.

**on** Enables verbose mode.

## Description

The output format set by this command remains in effect for the duration of the administrative command-line interface (CLI) session or until the options are reset either by using this command or by specifying a output-format parameter as part of a command.

Running this command with no parameters displays the current output settings in the default output format, for example:

Paging	Rows	Format	Header	Verbose
off	-	default	on	off

**Note:** The output formats do not apply to help pages.

### Example

**Set the output format** The following example sets the output format to display in tabular form using a comma as the delimiter without header information:

```
admin_cli_utility>setoutput -fmt delim , -hdr off
admin_cli_utility>lspool -l -type default
DEFAULT,Default,10000,2500,25,80,10,64,Default Storage Pool
```

## tankpasswd

Generates a password file that enables you to log in to the administrative command-line interface, also known as sfscli.

►►—tankpasswd— -u —user\_name— -p —password—◄◄

### Parameters

- u** *user\_name*  
Specifies the user name.
- p** *password*  
Specifies the password associated with the given user name.

### Description

This command must be run before you attempt to log on to the administrative CLI for the first time and when your password changes.

This command creates a password file, named tank.passwd, in your home directory. By default, the sfscli utility expects this file to be in your home directory. You can change this location by modifying the SFS\_CLI\_PASSWDFILE environment variable.

**Note:** You must move the password file to the location expected by the sfscli utility.

The user ID and password that you specify in this command must be the same user ID and password that is specified in the LDAP server by your system administrator. The LDAP server is used to connect to the administrative agent, which authorizes or denies access to SAN File System each time you attempt to use the sfscli utility.

### Example

**Generate the password file** The following example generates a password file for user saki:

```
$ pwd
/home/saki
$ tankpasswd -u saki -p mypassword
The password file was successfully written to: /home/saki/.tank.passwd
```



---

## Administrative commands

The following table provides a brief description and administrative role for each command in the administrative command-line interface.

Command	Description	Roles
<b>General</b>		
"exit" on page 125	Ends an stsccli session.	Monitor, Backup, Operator, Administrator
"help" on page 126	Displays a list of commands available in the administrative command-line interface and optionally displays the syntax or brief description of each command.	Monitor, Backup, Operator, Administrator
"quit" on page 127	Ends the stsccli session.	Monitor, Backup, Operator, Administrator
"setoutput" on page 127	Sets the output format for the administrative command-line interface.	Monitor, Backup, Operator, Administrator
<b>Alerts and logs</b>		
"addsnmpmgr" on page 137	Adds an SNMP manager to receive SNMP traps. To generate traps, you must add an SNMP manager and set the SNMP traps to be generated (using the <b>settrap</b> command).	Administrator
"catlog" on page 142	Displays the contents of the various log files maintained by the administrative server and the cluster.	Monitor, Backup, Operator, Administrator
"clearlog" on page 154	Clears the audit log and cluster log files. Because the event log is a subset of the cluster log, the event log is also cleared when you clear the cluster log.	Administrator
"collectdiag" on page 155	Collects data on all SAN File System components running in the system and saves the diagnostic results to various files on one or more specified engines.	Operator, Administrator
"lssnmpmgr" on page 201	Displays a list of SNMP managers and their attributes.	Monitor, Backup, Operator, Administrator
"lstrapsetting" on page 201	Displays a list of event types that currently generate an SNMP trap.	Monitor, Backup, Operator, Administrator
"rmsnmpmgr" on page 227	Removes an SNMP manager (recipient).	Administrator
"settrap" on page 236	Specifies whether an SNMP trap is generated and sent to all SNMP managers when a specific type of event occurs on the metadata server.	Administrator
<b>Clients</b>		
"addprivclient" on page 136	Grants privileged access to the SAN File System global namespace to the specified clients.	Administrator

Command	Description	Roles
"lsclient" on page 173	Displays a list of clients that are currently being served by one or more metadata servers in the cluster.	Monitor, Backup, Operator, Administrator
"reportclient" on page 216	Displays a list of clients that have access to the specified volume or LUN.	Monitor, Backup, Operator, Administrator
"rmprivclient" on page 226	Revokes privileged access to the SAN File System global namespace from the specified clients.	Administrator
<b>Cluster</b>		
"chadvclusterconfig" on page 144	Modifies the advanced cluster settings that require a restart when changed. <b>Note:</b> This command should be used only by trained service personnel.	Administrator
"chclusterconfig" on page 146	Modifies the cluster settings that do not require a restart when changed.	Administrator
"quiescecluster" on page 214	Changes the state of all metadata servers in the cluster to one of three quiescent states.	Operator, Administrator
"resumecluster" on page 220	Brings all metadata servers in the cluster to the online state.	Operator, Administrator
"startcluster" on page 239	Starts all metadata servers in the cluster and brings them to the full online state.	Operator, Administrator
"statcluster" on page 243	Displays status, network, workload, and configuration information about the cluster.	Monitor, Backup, Operator, Administrator
"stopcluster" on page 258	Stops all metadata servers in the cluster gracefully.	Operator, Administrator
"upgradecluster" on page 262	Verifies that all the engines in the cluster have been upgraded to the new version of the software and then initiates the cluster upgrade process.	Administrator
<b>Engines</b>		
"lsengine" on page 178	Displays a list of storage engines and their attributes.	Monitor, Backup, Operator, Administrator
"restartengine" on page 219	Reboots one or more storage engines.	Operator, Administrator
"startengine" on page 240	Starts the CPU and operating system on one or more storage engines.	Operator, Administrator
"statengine" on page 246	Displays status information about a specific storage engine.	Monitor, Backup, Operator, Administrator
"stopengine" on page 259	Shuts down the operating system and powers off one or more storage engines.	Operator, Administrator
<b>Filesets</b>		
"attachfileset" on page 138	Attaches an existing detached fileset to a specific point in the global namespace or moves an already attached fileset to a new attach location.	Administrator

Command	Description	Roles
"autofilesetserver" on page 140	Reassigns the specified filesets to be hosted by another metadata server automatically if the current metadata server goes down.	Administrator
"chfileset" on page 148	Modifies the settings of one or more filesets.	Administrator
"detachfileset" on page 156	Detaches one or more filesets from the global namespace.	Administrator
"lsfileset" on page 181	Displays a list of filesets and their attributes.	Monitor, Backup, Operator, Administrator
"mkfileset" on page 205	Creates a new fileset.	Administrator
"reportfilesetuse" on page 216	Displays the filesets usage statistics per storage pool.	Monitor, Backup, Operator, Administrator
"rmfileset" on page 222	Removes one or more empty, detached filesets and optionally the files in the filesets, including any FlashCopy images.	Administrator
"setfilesetserver" on page 231	Reassigns an existing fileset to be hosted by a different metadata server.	Administrator
"statfileset" on page 252	Displays the transaction statistics for one or more specified filesets being served by the local metadata server.	Monitor, Backup, Operator, Administrator
<b>FlashCopy images</b>		
"lsimage" on page 184	Displays a list of FlashCopy images in all filesets or a specified fileset.	Monitor, Backup, Operator, Administrator
"mkimage" on page 208	Creates a near-instantaneous FlashCopy image of the file layout and contents of the specified fileset and stores it in that fileset.	Backup, Operator, Administrator
"reverttoimage" on page 220	Reverts the current fileset to a specified FlashCopy image of the file layout and contents.	Administrator
"rmimage" on page 223	Deletes one or more FlashCopy images for a specific fileset.	Backup, Operator, Administrator
<b>LDAP</b>		
"chldapconfig" on page 150	Modifies LDAP configuration settings.	Administrator
"statldap" on page 253	Display a list of LDAP settings.	Monitor, Backup, Operator, Administrator
<b>Metadata</b>		
"buildrrscript" on page 141	Converts the system-metadata disaster-recovery file, created by the <b>mkdrfile</b> command, into a set of recovery scripts. You then run these scripts to recreate the system metadata in the event of a disaster.	Backup, Operator, Administrator
"lsdrfile" on page 176	Displays a list of system-metadata disaster-recovery files.	Monitor, Backup, Operator, Administrator

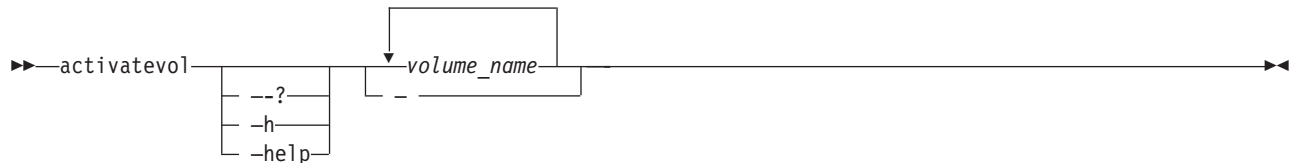
Command	Description	Roles
"mkdrfile" on page 207	Creates a new system-metadata disaster-recovery dump file.	Backup, Operator, Administrator
"rmdrfile" on page 223	Deletes an existing system-metadata disaster-recovery dump file.	Backup, Operator, Administrator
"startmetadatabackup" on page 240	Starts the utility that performs a consistency check on the metadata for the entire system or a set of filesets, generates reports in the cluster log, and optionally repairs inconsistencies in the metadata.	Administrator
"stopmetadatabackup" on page 260	Stops the metadata check utility that is currently in progress.	Administrator
<b>Metadata servers</b>		
"addserver" on page 137	Adds a metadata server on a specific engine to the cluster.	Administrator
"dropserver" on page 158	Stops and drops a metadata server on a specific engine from the cluster.	Administrator
"lsautorestart" on page 169	Displays a list of metadata servers and the automatic-restart settings for each.	Monitor, Backup, Operator, Administrator
"lsserver" on page 197	Displays a lists of all metadata servers in the cluster and their attributes if issued from the master metadata server, or displays attributes about the local metadata server if issued from a subordinate metadata server.	Monitor, Backup, Operator, Administrator
"setautorestart" on page 229	Sets the configuration parameters for the metadata server restart service.	Administrator
"setport" on page 235	Sets the port values for the metadata server.	Administrator
"startautorestart" on page 238	Enables the metadata server to restart automatically if it is down.	Administrator
"startserver" on page 242	Starts the specified metadata server.	Operator, Administrator
"statserver" on page 256	Displays status, configuration, and workload information for a specific metadata server in the cluster, if issued from the master metadata server. Lists status, configuration, and workload information for the local metadata server, if issued from a subordinate metadata server.	Monitor, Backup, Operator, Administrator
"stopautorestart" on page 258	Disables the metadata server from restart automatically if it is down.	Administrator
"stopserver" on page 261	Shuts down a subordinate metadata server gracefully.	Operator, Administrator
<b>Policies</b>		
"catpolicy" on page 144	Displays the rules of the specified policies.	Monitor, Backup, Operator, Administrator
"lspolicy" on page 190	Lists the active and inactive policies.	Monitor, Backup, Operator, Administrator

Command	Description	Roles
"mkpolicy" on page 209	Creates a policy.	Administrator
"rmpolicy" on page 224	Deletes one or more inactive policies.	Administrator
"statpolicy" on page 254	Displays policy statistics for each fileset, optionally based on a specified storage pool or rule.	Monitor, Backup, Operator, Administrator
"usepolicy" on page 263	Directs the metadata server to make an existing policy the active policy and applies its rules to all subsequent file creations.	Administrator
<b>Processes</b>		
"lsproc" on page 195	Displays a list of long-running processes that are not yet complete and their attributes.	Monitor, Backup, Operator, Administrator
<b>Storage pools</b>		
"chpool" on page 152	Modifies the settings of one or more storage pools.	Administrator
"disabledefaultpool" on page 157	Disables the use of the default storage pool and makes the current default storage pool a regular storage pool.	Administrator
"lspool" on page 192	Displays a list of the existing storage pools and their attributes.	Monitor, Backup, Operator, Administrator
"mkpool" on page 211	Defines one or more new user storage pools.	Administrator
"rmpool" on page 225	Deletes one or more empty, unreferenced storage pools.	Administrator
"setdefaultpool" on page 230	Changes a user storage pool to the default storage pool, and changes the previous default storage pool to a regular user storage pool.	Administrator
<b>Users</b>		
"lsadmuser" on page 167	Displays a list of administrative users and their attributes.	Monitor, Backup, Operator, Administrator
"resetadmuser" on page 218	Forces all administrative users to log in again.	Administrator
<b>Volumes and LUNs</b>		
"activatevol" on page 135	Expands the size of one or more volumes.	Administrator
"chvol" on page 153	Modifies the settings of one or more volumes.	Administrator
"expandvol" on page 159	Modifies the settings of one or more volumes.	Administrator
"lslun" on page 187	Lists the LUNs that are accessible from SAN File System.	Monitor, Backup, Operator, Administrator

Command	Description	Roles
"lsvol" on page 202	Displays a list of available volumes.	Monitor, Backup, Operator, Administrator
"mkvol" on page 212	Adds one or more volumes to a storage pool.	Administrator
"rediscoverluns" on page 215	Rediscovered LUNs on the SAN that are visible to SAN File System.	Monitor, Backup, Operator, Administrator
"reportvolfiles" on page 217	Displays a list of files, and their attributes, for the specified volume.	Backup, Operator, Administrator
"rmvol" on page 227	Removes one or more volumes from a storage pool and redistributes the contents to other volumes in the same storage pool.	Administrator
"suspendvol" on page 262	Suspends one or more volumes so that the metadata server cannot allocate new data on the volumes.	Administrator

## activatevol

Activates a suspended volume so that the metadata server can allocate new data on the volume.



### Parameters

#### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### *volume\_name*

Specifies the names of the one or more volumes to activate.

- Specifies that you want to read the names of one or more volumes to activate from stdin (for example, - << /work/vol\_list.txt).

### Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine that hosts the master metadata server to run this command.

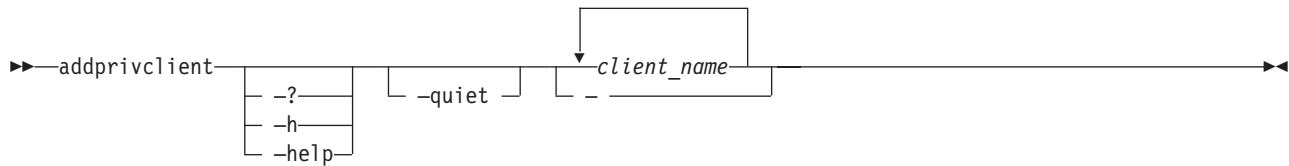
### Example

**Activate two volumes** The following example activates two volumes: vol1 and vol2.

```
sfsccli> activatevol vol1 vol2
Volume vol1 activated.
Volume vol2 activated.
```

## addprivclient

Grants privileged access to the SAN File System global namespace to the specified clients.



### Parameters

#### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### -quiet

Turns off confirmation messages for this command.

#### client\_name

Specifies one or more names of clients whose root user (on UNIX) or Administrator user (on Windows) is to be granted full, unrestricted access to the SAN File System global namespace. You can specify up to 30 client names, and the total characters in the client names and the commas between client names must not exceed 256 characters.

This command does not verify that the specified client names are valid because the metadata server does not keep static information about the clients. It does not keep a list of allowed or possible clients. It has only a list of clients that it allows privileged access if they present themselves. This command lists the clients that are currently accessing data.

- Specifies that you want to read the names of one or more clients from stdin (for example, - << /work/client\_list.txt).

### Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine that hosts the master metadata server to run this command.

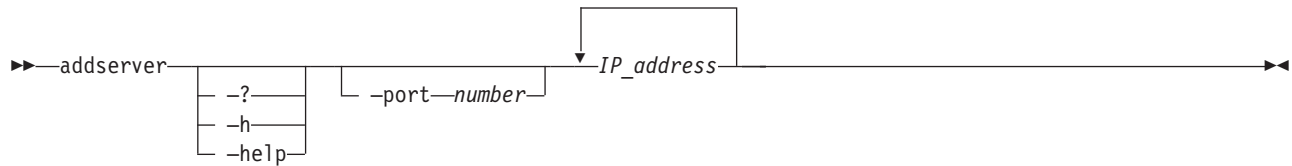
### Example

**Grant root privileges** This example grants root privileges to the client named client02:

```
sfsccli> addprivclient client02
Are you sure you want to add client02 as a privileged client? [y/n] Y
Privileged client access successfully granted for client02.
```

## addserver

Adds a metadata server on a specific engine to the cluster.



### Parameters

#### `-? | -h | -help`

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### `-port number`

Specifies the port number on the engine to which to add the metadata servers. The default value is the default cluster port, 1737.

#### `IP_address`

Specifies the Internet Protocol (IP) address corresponding to one or more metadata servers that you want to add the cluster.

### Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine that hosts the master metadata server to run this command.

### Description

Typically, you issue this command during set up. You must run this command from the engine that is running the master metadata server. After you add the metadata servers to the cluster, they can begin serving the workload.

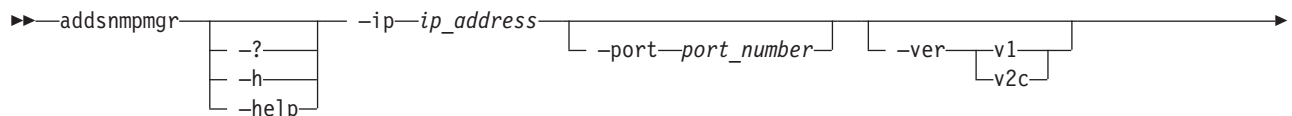
### Example

**Adds a metadata server to the cluster** The following example adds the metadata server 179.24.17.34 to the cluster.

```
sfsccli> addserver 179.24.17.34
Server 179.24.17.34 added to the cluster.
```

## addsnmpmgr

Adds an SNMP manager to receive SNMP traps.





## Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

**-ip** *ip\_address*

Specifies the IP address of the SNMP trap receiver.

**-port** *port\_number*

Specifies the SNMP-trap port number. The default port number is 162.

**-ver** *v1 | v2c*

Specifies the SNMP version. You can specify one of the following values:

**v1** Sends SNMP version 1 traps. This is the default value.

**v2c** Sends SNMP version 2 notifications using community-based security.

**-community** *community*

Specifies the community as any alphanumeric string used to authenticate v1 or v2c traps. The default community is “public.”

Commas and colons are not valid characters in the community name. This name can be no longer than 128 characters in length.

## Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

To generate traps, you must add an SNMP manager and set the SNMP traps to be generated (using the **settrap** command). If an SNMP manager is not added, traps are not generated.

You can add up to two SNMP managers to SAN File System. One of these SNMP managers can be used by the optional Call Home feature.

When you change a disruptive cluster setting, dynamic cluster settings (such as SNMP settings) cannot be modified until you reboot the cluster.

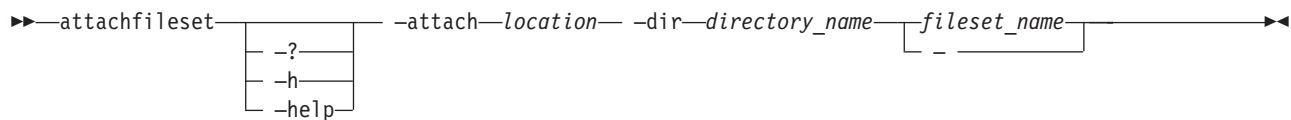
## Example

**Adds an SNMP manager** The following example adds an SNMP manager to the SNMPPMgr1 community using SNMP v2c.

```
sfscli> addsnmpmgr -ip 192.168.0.1 -port 8192 -ver v2c -community SNMPPMgr1
SNMP manager 192.168.0.1 successfully added.
```

## attachfileset

Attaches an existing detached fileset to a specific point in the global namespace or moves an already attached fileset to a new attach location.



## Parameters

### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### -attach *location*

Specifies the directory path at which to attach the fileset. The root fileset in the global namespace must be included in the directory path. For example, to attach the fileset to the /home location, specify `-attach sanfs/home`. To attach the fileset directly to the root of the global namespace, use `-attach sanfs`.

Use only forward slashes (/) in the directory path for delimiters.

This directory *must* already exist.

**Note:** The name of the root fileset is the same as the name of the cluster. This name is "sanfs."

### -dir *directory\_name*

Specifies the directory name for the root of the fileset directory tree. The directory name *must not* already exist. This directory can be up to 256 characters in length.

The directory must not contain backslash (\) or forward slash (/) characters as delimiters.

The attach point (*location/directory\_name*) appears as a drive and directory on clients for Windows or as a file system for client based on UNIX. It is recommended that you map the name of the directory to its associated fileset by using the same name as the fileset plus an additional suffix to distinguish it as the directory and not the fileset. For example, if you want to attach the fileset named work to the root of the global namespace, you would specify "sanfs" for the *location* and specify "work\_ap" for the *directory\_name*. On a client for Windows, the "work\_ap" directory would appear under the "sanfs" drive.

### *fileset\_name*

Specifies the name of the fileset to attach.

- Specifies that you want to read the name of the fileset to attach from stdin (for example, `- << /work/cnt_list.txt`).

## Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

You make a fileset part of the global namespace by attaching it to a specific location. To a client, this attach point looks like a regular directory.

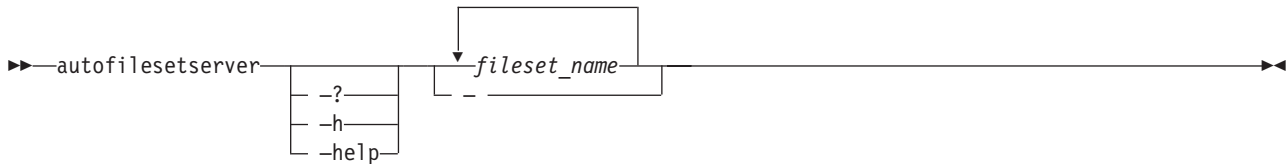
## Example

**Attaching a fileset** The following example attaches fileset `cntA` to the `sanfs/homes` location in the global namespace using `cntA_ap` as its directory name.

```
sfscli> attachfileset -attach sanfs/homes -dir cntA_ap cntA
Fileset cntA attached to directory sanfs/homes/cntA_ap.
```

## autofilesetserver

Reassigns the specified dynamically-assigned filesets to another metadata server by the system.



### Parameters

`-? | -h | -help`

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

*fileset\_name*

Specifies the names of one or more filesets to dynamically reassign to a new metadata server.

- Specifies that you want to read the names of one or more filesets to reassign from stdin (for example, `- << /work/cnt_list.txt`).

### Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

The current metadata server must be in one of the following states:

- Not running
- Offline
- Online

### Description

If the metadata server that is dynamically assigned to the filesets goes offline, the filesets are moved to another metadata server. When the assigned metadata server comes back online, the filesets *do not* fail back to the assigned metadata server.

## Example

**Reassign a fileset** The following example dynamically reassigns two filesets (`cnt_A` and `cnt_B`) to another metadata server:

```
sfsccli> autofilesetserver cnt_A cnt_B
Automatic server assignment for cnt_A is enabled.
Automatic server assignment for cnt_B is enabled.
```

## builddrscript

Converts the system-metadata disaster-recovery file into a set of recovery scripts used to recreate the system metadata in the event of a disaster.



### Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

*file\_name*

Specifies the name of an existing system-metadata disaster-recovery file.

### Prerequisites

You must have Backup, Operator, or Administrator privileges to use the command.

### Description

**Note:** This command operates only against the local engine.

This command is used in conjunction with the **mkdrfile** command to create a set of recovery scripts that are used to recreate the system metadata in the event of a disaster. The **builddrscript** command is run against the system-metadata disaster-recovery file created by the **mkdrfile** command.

**Note:** The output for this command is written to the `/usr/tank/server/DR` directory. This command will overwrite any files that were created by a previous run of this command. If you want to preserve the existing files, copy them to another directory.

You would run this command against the system-metadata disaster-recovery file only if you are intending to recreate the system metadata.

After this command generates the recovery scripts, you must edit the scripts before running them.

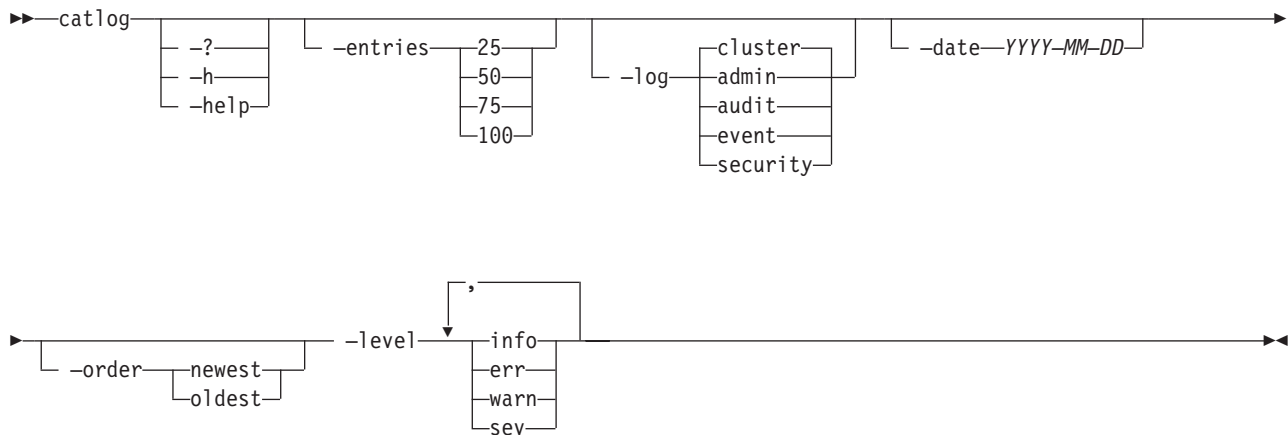
### Example

**Builds the recovery scripts** The following example builds the disaster recovery scripts from the system-metadata disaster-recovery file named `dr1`.

```
sfsccli> builddrscript dr1
Disaster recovery script files for "dr1" were built successfully.
```

## catlog

Displays the contents of the various log files maintained by the administrative server and the cluster.



### Parameters

#### `-? | -h | -help`

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### `-entries`

Specifies the number of log entries to show at a time, from oldest to newest. Valid values are 25, 50, 75, or 100. If not specified, this command shows the entire log.

#### `-log`

Displays entries in the specified log, ordered by timestamp starting with the most recent entry. The default is **cluster**.

**admin** Displays entries in the administrative log, which maintains a history of messages created by the administrative server.

**audit** Displays entries in the audit log, which maintains a history of all commands issued by any administrator for all metadata servers in the cluster.

#### **cluster**

Displays entries in the cluster log, which maintains a history of messages created by all metadata servers in the cluster.

**event** Displays event entries in the event log, which maintains a history of event messages issued by all metadata servers in the cluster.

#### **security**

Displays entries in the security log, which maintains a history of administrative-user login activity.

#### `-date`

Specifies the date at which you want the displayed log entries to start. The date must be in the format YYYY-MM-DD, where YYYY is the year, MM is the month, and DD is the day.

This date must be the current date or older. Future dates are not acceptable.

**-order**

Specifies the direction of the displayed log entries. You can specify one of the following values:

**newest**

Displays the log entries form newest to oldest. This is the default value if the **-date** parameter is not specified.

**oldest** Displays the log entries form oldest to newest. This is the default value if the **-date** parameter is specified.

**-level info | err | warn | sev**

Specifies the severity level of the displayed log entries. If not specified, all severity levels are displayed. You can specify one or more levels, separate by a comma and no space (for example, **-level info,warn,err**)

**Description**

If you run this command from an engine hosting a subordinate metadata server, logs for only the local engine are displayed. If you run this command from the engine hosting the master metadata server, logs for the entire cluster are displayed.

If there are log entries that have not been displayed, you are prompted to press Enter to display the next set of entries or to type **exit** and press Enter to stop.

This command displays the following information for the specified log:

- Message identifier.
- Severity level (Info, Error, Warning, Severe).
- Message type (Normal or Event).
- Name of the metadata server that generated the message.
- Date and time the message was generated.
- Message description.

**Tip:** It is important that the date and time are correct on the metadata servers so messages are logged correctly and log entries are displayed correctly when using the **-time** parameter.

**Example**

**Display the event entries in the cluster log** The following example displays the error messages in the event log that occurred on or after January 4, 2003.

```
sfscli> catlog -log event -date 2003-01-04 -level err
```

ID	Level	Type	Server	Date and Time
HSTSS0009E	Error	Event	ST3	Jan 2, 2003 8:39:15 PM
HSTNL0019E	Error	Event	ST2	Jan 2, 2003 8:40:46 PM

**Message**

```
=====
The Metadata server rpm is not installed.
Unable to extract boot record when server is running.
```

## catpolicy

Displays the rules of the specified policies.



### Parameters

#### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### *policy\_name*

Specifies the names of one or more policies to display.

- Specifies that you want to read the names of one or more policies to display from stdin (for example, - << /work/policies\_list.txt).

### Prerequisites

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

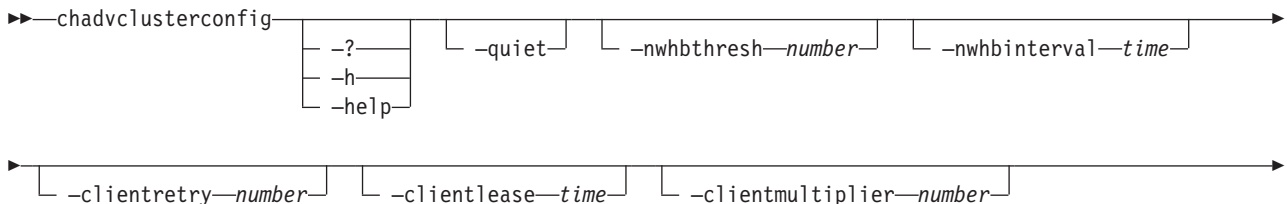
### Example

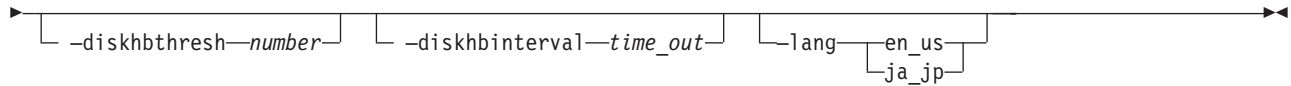
**Display the rules in a policy** The following example displays the rules for the DEFAULT and the ActivePolicy policies

```
sfsccli> catpolicy DEFAULT_POLICY activePolicy
DEFAULT_POLICY:
 VERSION 1
 /* Default Policy Set
 Assign all files to default storage pool.
 When no rule applies to a file, default storage pool is assigned.
 */
activePolicy:
 VERSION 1 /* Do not remove or change this line!*/
 rule 'stgRule1' set stgpool 'pool1' for FILESET ('cnt_A')
 rule 'stgRule2' set stgpool 'pool2' where NAME like '%.doc'
 rule 'stgRule3' set stgpool 'pool3' where DAYOFWEEK(CREATION_DATE) == 1
 rule 'stgRule4' set stgpool 'pool4' where USER_ID <= 100
```

## chadvclusterconfig

Modifies the advanced cluster settings.





## Parameters

### **-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### **-quiet**

Turns off confirmation messages for this command.

### **-nwhbthresh** *number*

Specifies the maximum number of missed network heartbeats before the metadata server declares a possible path-failure problem with another metadata server in the cluster. If this threshold is reached or exceeded and the metadata server is running, the metadata server goes into the not-running state, and stays in that state until it is repaired. If the metadata server was not running, the ejected metadata server abnormally ends, restarts, and rejoins the cluster.

You can specify a value from 1 to 100. The default value is 40.

### **-nwhbinterval** *time*

Specifies the network heartbeat interval, in milliseconds, for periodic pings among metadata servers in the cluster. You can specify a value from 200 to 10 000. The default value is 500.

### **-clientretry** *number*

Specifies the maximum number of times a send can be attempted from the metadata server to the client before declaring the client's session lease to be expired. You can specify a value from 1 to 100. The default value is 5.

### **-clientlease** *time*

Specifies the amount of time, in seconds, that a lock is leased to a client when the metadata server grants a lock. The metadata server applies the client-lease-period multiplier to this value before ending the lease. You can specify a value from 10 to 120. The default value is 20.

### **-clientmultiplier** *number*

Specifies the client-lease-period multiplier. This multiplier accommodates possible networking delays between the time a message gets sent from the client until it is received by the metadata server. You can specify a value from 0 to 4. A value of 0 prevents this multiplier from affecting the client-lease period. The default value is 2.

### **-diskhbthresh** *number*

Specifies the maximum number of missed disk heartbeats before the metadata server is dropped from the cluster. You can specify a value from 1 to 100. The default value is 4.

### **-diskhbinterval** *time*

Specifies the disk heartbeat interval, in milliseconds, for periodic pings among metadata servers in the cluster. You can specify a value from 200 to 10 000. The default value is 500.

### **-lang**

Specifies the language that the metadata server runs. You can specify one of these values:



ja\_jp Japanese UTF8

### Prerequisites

You must have Administrator privileges to use the command.

### Description

After you run this command, you cannot run the **chadvclusterconfig** or **chclusterconfig** command until you restart the cluster.

**Restriction:** This command cannot be used when the cluster is quiesced.

**Changes the network heartbeat threshold** The following example changes the network heartbeat threshold to 10 missed heartbeats.

## chclusterconfig

```

graph LR
 subgraph Section1 []
 direction TB
 S1_1["-?"]
 S1_2["-h"]
 S1_3["-help"]
 S1_4["-quiet"]
 S1_5["-spacereclaim-minutes"]
 S1_6["-mstrbuff-size"]
 end
 subgraph Section2 []
 direction TB
 S2_1["-subbuff-size"]
 S2_2["-admproc-number"]
 S2_3["-serverproc-number"]
 S2_4["-privclient-list"]
 end
 subgraph Section3 []
 direction TB
 S3_1["-rsauser-rsa user"]
 S3_2["-rsapasswd"]
 end
 Section1 --- Section2 --- Section3

```

## Parameters

### **-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### **-quiet**

Turns off confirmation messages for this command.

### **-spaceclaim *minutes***

Specifies the storage pool-space-reclamation interval, in minutes. You can specify a value from 0 to 1 440. The default is 60. A value of 0 disables storage pool-space reclamation.

### **-mstrbuff *size***

Specifies the size, in 4-KB pages, of the master metadata server buffer. You can specify a value from 2 048 and 8 192. The default is 2 048.

**Note:** The buffer size can only be increased; it cannot be decreased.

### **-subbuff *size***

Specifies the size, in 4-KB pages, of the subordinate metadata server buffer. You can specify a value from 30 000 and 250 000. The default is 200 000.

**Restriction:** The buffer size can only be increased; it cannot be decreased.

### **-admproc *number***

Specifies the maximum limit for administrative processes, including the number of metadata server commands (from the administrative CLI or SAN File System console), that can run simultaneously. This limit is independent of the server workload process limit. A few commands, such as those that change the state of the cluster, can run on a temporary process even if the limit is currently exceeded. You can specify a value from 4 to 10. If not specified, the default value is 4.

**Restriction:** This limit can only be increased; it cannot be decreased.

### **-serverproc *number***

Specifies the maximum limit for the metadata server workload process, which involves only metadata server workload operations. You can specify a value from 10 to 50. The default value is 20.

**Restriction:** This limit can only be increased; it cannot be decreased.

### **-privclient *list***

Specifies a list of clients, separated by commas, that are to have root-level privileges (for example, `-privclient saki,mefi,leki`). The current list is replaced with the new list of clients. You can specify from zero to 30 clients; each client name can be up to 256 characters in length and is case-sensitive.

**Restriction:** This new list replaces the entire list of current privileged clients. If there are clients that have root or administrative privileges and you want to add one more client, you must specify both the current and new clients in the new list.

To remove all clients with root or administrative privileges, specify **-privclient ""**; however, at least one privileged client is required to create files and directories under the root directory of filesets.

This command does not verify that the specified client names are valid because the metadata server does not keep static information about the clients. It does not keep a list of allowed or possible clients. It has only a list of clients that it allows privileged access if they present themselves. This command lists the clients that are currently accessing data.

**-rsauser** *rsa\_user*

Specifies the Remote Service Adapter (RSA) user ID that is used to access information from the engines. The RSA user ID and password must be the same for all engines that host the cluster of metadata servers. Also, the values must match those in the RSA system.

**Attention:** Verify that the user ID is set to the correct ID. If you set an incorrect RSA user ID, all RSA features (including all engine commands) become unusable.

**-rsapasswd**

Specifies the password corresponding to the RSA user ID. You will be prompted to enter the password and then confirm it.

**Attention:** Verify that the password is set to the correct password. If you set an incorrect RSA password, all RSA features (including all engine commands) become unusable.

### Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine that hosts the master metadata server to run this command.

### Description

You might have to restart the cluster for changes to take effect.

**Restriction:** If you modify any disruptive settings using this command or the **chadvclusterconfig** command, you cannot modify any cluster settings until you restart the cluster.

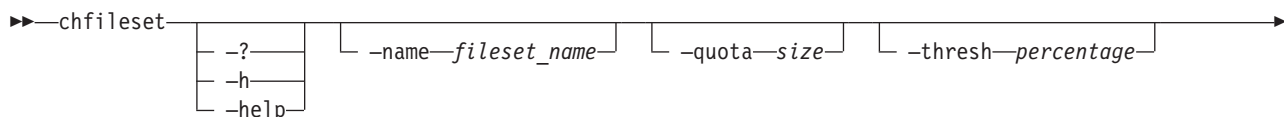
### Example

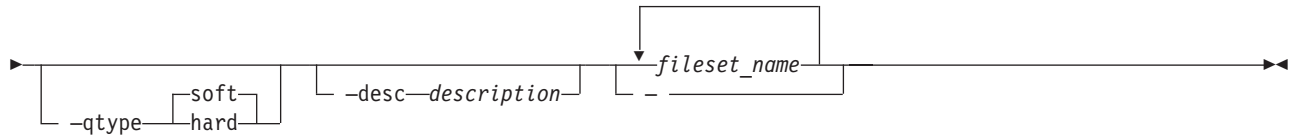
**Changes the cluster settings** The following example changes the subordinate metadata server buffer size to 250000 and changes RSA password for the cluster:

```
sfscli> chclusterconfig -subbuff 250000 -rsapasswd
Are you sure you want to change cluster configuration settings? [y/n] Y
Enter new RSA password:
Confirm new RSA password:
The cluster was modified successfully.
```

## chfileset

Modifies the settings of one or more filesets.





## Parameters

### **-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### **-name**

Specifies the name to assign to the fileset. This name can be up to 256 characters in length.

### **-quota *size***

Specifies the maximum size limit, in megabytes, for the specified fileset that, when exceeded, causes the metadata server to generate an alert. You can specify a number from 0 to 1 073 741 824 MB (1 PB). If set to 0, there is no quota size limit for this fileset, and alerts are not sent.

**Note:** If a hard quota already exists, you must not set the new size to a value less than the current quota size for this fileset.

### **-thresh *percentage***

Specifies the maximum percentage (alert threshold) of the specified quota size for this fileset that, when exceeded, causes the metadata server to generate an alert. You can specify a value between 0 and 100. If set to 0, no alerts are generated.

### **-qtype **hard** | **soft****

Specifies the quota type for the fileset. You can specify one of the following values:

**hard** A hard quota produces a log message and potential alert when the quota is met, and denies requests for additional space.

**soft** A soft quota produces a log message and potential alert when the quota size is exceeded, but grants requests space allocation.

### **-desc *description***

Specifies a description for the fileset. The description must be enclosed in matching single (') or double (") quotation marks if it contains any blank characters. This description can be up to 256 characters in length.

The default is an empty string.

### ***fileset\_name***

Specifies the names of one or more filesets to modify.

**Note:** If you specify a name using the **-name** option, you may specify one only fileset.

- Specifies that you want to read the names of one or more filesets to modify from stdin (for example, - << /work/cnt\_list.txt).

## Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

You cannot modify the name for multiple filesets at one time.

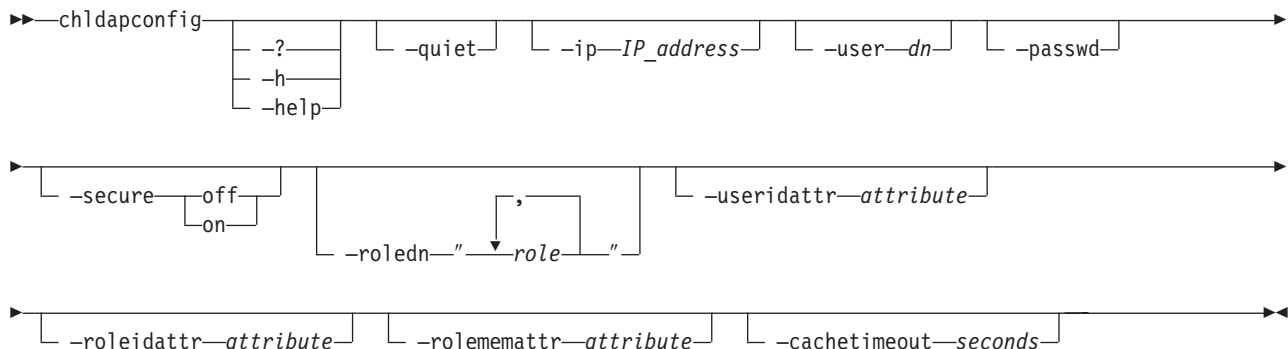
## Example

**Modify parameters of two filesets** The following example changes the quota and threshold values for two filesets (*cnt\_A* and *cnt\_B*). It, sets the quota to 1 000 MB and specifies to send an alert when the quota reaches 70%.

```
sfscli> chfileset -quota 1000 -thresh 70 -qtype soft cnt_A cnt_B
Fileset cnt_A modified.
Fileset cnt_B modified.
```

## chldapconfig

Modifies LDAP configuration settings on the metadata server.



## Parameters

### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### -quiet

Turns off confirmation messages for this command.

### -ip IP\_address

Specifies the IP address of the LDAP server.

### -user dn

Specifies the administrative LDAP user ID used to make authentication queries. This is the user ID that is used to sign into the LDAP server. This value does not represent the user IDs that have been created for SAN File System.

### -passwd

Specifies the password corresponding to the administrative LDAP user ID. If you specify this parameter, you are prompted to supply the password and then confirm it.

**-secure off | on**

Specifies whether the LDAP connection is made using Secure Sockets Layer (SSL). You can specify one of the following values:

**off** The LDAP connection uses a standard, insecure socket.

**on** The LDAP connection uses Secure Sockets Layer (SSL). This is the initial value.

**-roledn "roles"**

Specifies an LDAP distinguished name (DN) that points to where the role definitions are located in the LDAP server. Within a distinguished name, you can specify one or more roles, separate by a comma. If you include spaces in this list, you must enclose the list in double quotation marks, for example, "ou=Roles,o=IBM,c=US".

To clear this value, specify empty double quotation marks (for example "").

**Tip:** If you clear this field, the administrative functions might become unusable.

The maximum length for the DN (the entire list of roles) is 256 characters.

**-useridattr attribute**

Specifies the LDAP attribute containing the login name. The initial value is **uid**. For Active Directory, the recommended value is **sAMAccountName**.

**-roleidattr attribute**

Specifies the LDAP attribute containing the name of the role. The initial value is **cn**. For Active Directory, the recommended value is **description**.

**-rolememattr attribute**

Specifies the LDAP attribute containing the members of the role. The initial value is **roleOccupant**. For Active Directory, the recommended value is **member**.

**-cachetimeout seconds**

Specifies the maximum age, in seconds, of items in LDAP cache. The initial value is 600.

The items in the LDAP cache are removed if the cache timeout is reached. You are then authenticated in the LDAP server rather than in a cache entry the next time you attempt to use the administrative functions.

## Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command

## Description

**Attention:** This command only changes settings on the metadata servers. It does not modify settings on the LDAP server. The settings on the metadata server must match the settings on the LDAP server exactly.

**Restriction:** You must change LDAP settings on the metadata server using a single **chldapconfig** command and then modify the same settings on the LDAP server, before issuing any other administrative commands. If you modify settings on the LDAP server first, the user authentication fails, and you will not be able to issue administrative commands.

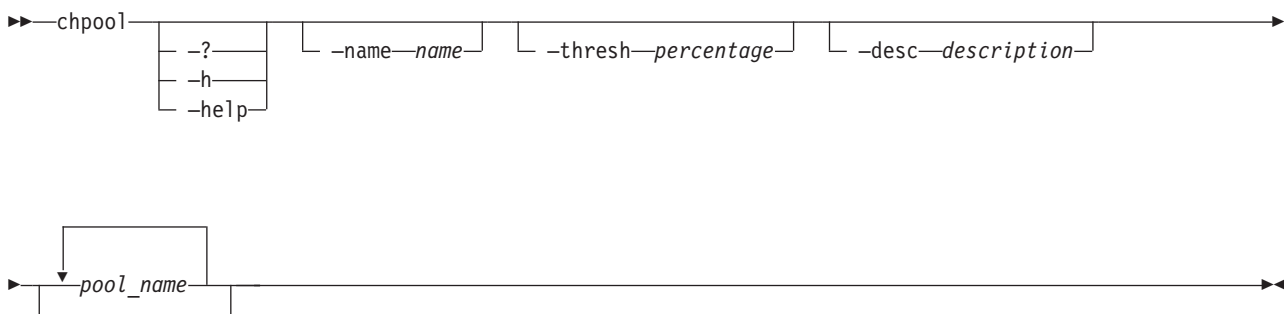
## Example

**Modify LDAP settings** The following example modifies the user ID, password, and base distinguished names (DN) on the metadata server to match the configuration on the LDAP server:

```
#sfscli chldapconfig -user cn=Administrator,dc=company,dc=com
-roledn "cn = foo, ou = bar, c = us" -passwd
Are you sure you want to change LDAP configuration settings? Administrative
interfaces will not be usable until the LDAP server is modified to match.
[y/n] Y
Enter new LDAP password:
Confirm new LDAP password:
The LDAP configuration was modified successfully.
```

## chpool

Modifies the settings of one or more storage pools.



### Parameters

#### **-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### **-name *name***

Specifies the name to assign to the storage pool. This name can be up to 256 characters in length.

**Note:** You cannot change the name of the SYSTEM storage pool.

#### **-thresh *percentage***

Specifies a percentage of the storage pool's estimated capacity that, when reached or exceeded, causes the metadata server to generate an alert. If the capacity drops and then reaches the percentage again, additional alerts are sent. You can specify a value between 0 and 100. If set to 0, no alert is generated.

#### **-desc *description***

Specifies a description for the storage pool. The description must be enclosed in matching single (') or double (") quotation marks if it contains any blank characters. This description can be up to 256 characters in length.

The default is an empty string.

**Note:** You cannot change the description of the SYSTEM storage pool.

*pool\_name*

Specifies the names of one or more storage pools to modify.

**Note:** If you specify a name using the **-name** parameter, you may specify only one storage pool.

- Specifies that you want to read the names of one or more storage pools to modify from stdin (for example, - << /work/stgpools\_list.txt).

### Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

### Description

You cannot modify the names of multiple storage pools at one time.

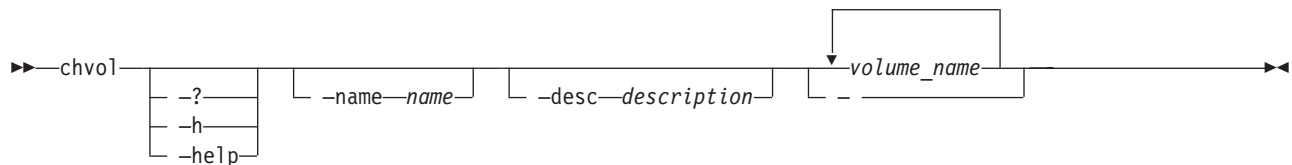
### Example

**Modify settings for two storage pools** The following example modifies the threshold setting for two storage pools: stgpools1 and stgpools2.

```
sfscli> chpool -thresh 70 stgpools1 stgpools2
Storage pool stgpools1 modified.
Storage pool stgpools2 modified.
```

## chvol

Modifies the settings of one or more volumes.



### Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

**-name *name***

Specifies the name to assign to the volume. The volume name must be unique within a cluster and can be up to 256 characters in length.

**-desc *description***

Provides a description of the volume. The description must be enclosed in matching single (') or double (") quotation marks if it contains any blank characters and can be up to 256 characters in length.

The default is an empty string.

*volume\_name*

Specifies the name of one or more volumes to modify.



**Note:** If you specify a name using the **-name** option, you may specify one only volume.

- Specifies that you want to read the names of one or more volumes to modify from stdin (for example, - << /work/vol\_list.txt).

### Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

### Description

You cannot modify the name of multiple volumes at one time.

### Example

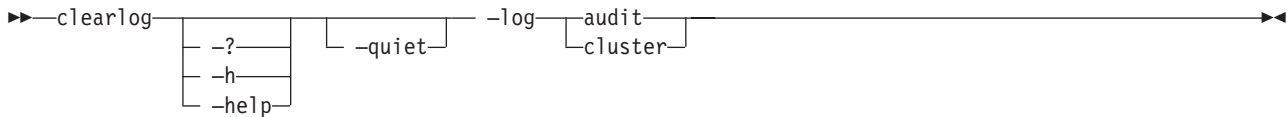
**Modify settings for multiple volumes** The following example modifies the description of all the volumes listed in the file */tmp/vol\_list.txt* to be "Engineering Volumes":

```
sfscli> chvol -desc "My SAN File System volumes" - << /tmp/vol_list.txt
```

```
Volume vol1 modified.
Volume vol2 modified.
Volume vol3 modified.
```

## clearlog

Clears the audit log and cluster log files.



### Parameters

#### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### -quiet

Turns off confirmation messages for this command.

#### -log

Specifies the log file to clear.

**audit** Clears all entries in the audit log, which maintains a history of messages created by the administrative server.

#### cluster

Clears all entries in the cluster log, which maintains a history of all messages issued by any administrator for all metadata servers in the cluster.

## Prerequisites

You must have Administrator privileges to use the command.

## Description

### Note:

1. If you run this command from an engine hosting a subordinate metadata server, the audit log and cluster log on only the local engine are cleared. If you run this command from the engine hosting the master metadata server, the audit logs and cluster logs on all engines are cleared.
2. Because the event log is a subset of the cluster log, the event log is also cleared when you clear the cluster log.

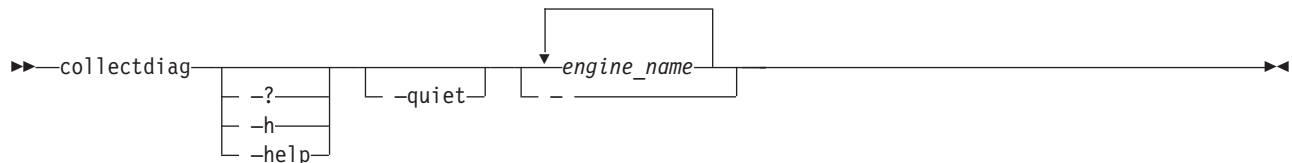
## Example

**Clear the cluster log** The following example clears the cluster log entries.

```
sfscli> clearlog -log cluster
Are you sure you want to remove all entries in the cluster log? [y/n] Y
Cluster log cleared.
```

## collectdiag

Collects data on all SAN File System components running in the system and saves the diagnostic results to various files on the local file system of each specified engine. This command is used to assist in problem determination.



## Parameters

### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### -quiet

Turns off confirmation messages for this command.

### engine\_name

Specifies the names of one or more engines for which to collect diagnostic data.

**Note:** You must be logged on to the master metadata server to issue this command against one or more non-local engine. If you are not logged on to the master metadata server, this command collects diagnostics data only for the local engine.

- Specifies that you want to read the name of the engine for which to gather diagnostic data from stdin (for example, `- << /work/vol_list.txt`).

## Prerequisites

You must have Operator or Administrator privileges to use the command.

## Description

Diagnostic data is saved on each specified engine in a subdirectory of the directory specified by the TANKDIR environment variable. The default directory is /usr/tank/OBDC. The subdirectory name is the timestamp when the data is collected (for example, /usr/tank/OBDC/20030507\_081010).

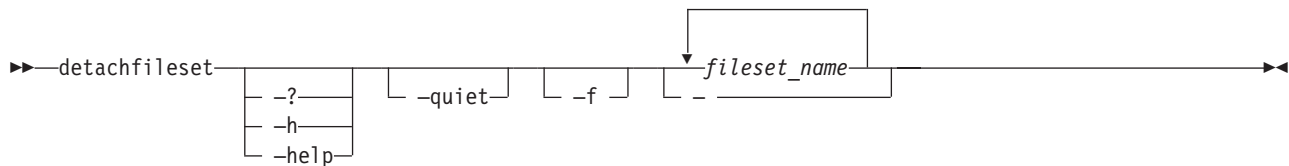
## Example

**Collect diagnostic data** The following example collects diagnostic data for engine ST1:

```
sfscli> collectdiag ST1
Are you sure you want to collect diagnostic data for ST1 since this
procedure may take a few minutes to complete? [y/n] Y
Diagnostic file successfully created for Engine ST1. Tip: See
usr/tank/pmf to read the diagnostic files.
```

## detachfileset

Detaches one or more filesets from the global namespace.



## Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

**-quiet**

Turns off confirmation messages for this command.

**-f**

Forces the metadata server to continue the detach operation even if there are open files in the fileset and suppresses any warning messages.

*fileset\_name*

Specifies the names of one or more filesets to detach.

**-**

Specifies that you want to read the names of one or more filesets to detach from stdin (for example, `- << /work/cnt_list.txt`).

## Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.



## Example

**Disable the default storage pool** The following example disables the default storage pool:

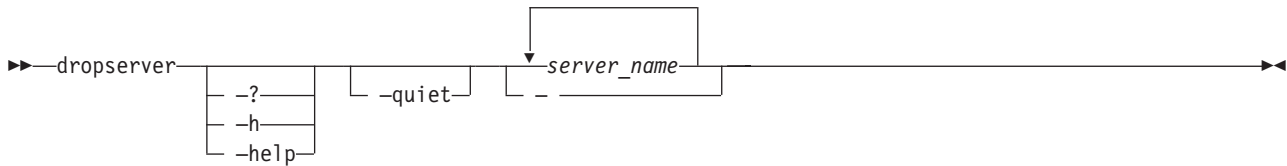
```
sfsccli> disabledefaultpool
```

Are you sure you want to disable the default storage pool? [y/n] Y

The default storage pool is now disabled. Files must match a policy rule to be created or saved.

## dropserver

Stops and drops a metadata server on a specific engine from the cluster.



### Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

**-quiet**

Turns off confirmation messages for this command.

*server\_name*

Specifies one or more metadata servers to drop from the cluster.

- Specifies that you want to read the names of one or more metadata servers to drop from stdin (for example, - << /work/cnt\_list.txt).

### Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

### Description

This command is typically issued when changing the IP address or upgrading the metadata server.

You cannot drop the last metadata server in the cluster.

## Example

**Drops a metadata servers from the cluster** The following example drops the metadata server ST4 from the cluster:

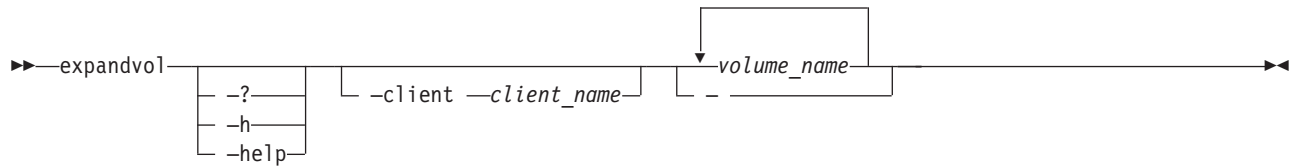
```
sfsccli> dropserver ST4
```

Are you sure you want to drop server ST4? Filesets manually assigned to the server will be reassigned to the remaining servers. [y/n] Y

Server ST4 successfully dropped from the cluster.

# expandvol

Expands the size of one or more specified volumes.



## Parameters

### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### -client *client\_name*

Specifies the name of a client that is zoned to access the volume that is being expanded.

To create a volume in a user storage pool, you must specify a SAN File System client that meets the following conditions:

- The client must have access to the volume that is being added as a SAN File System volume. To list all active clients that can access the volume, use the **reportclient -lun** command.
- The client must be active and appear in the registered (active) client list.

**Restriction:** Do not specify a client if you are expanding a volume in the system storage pool.

### *volume\_name*

Specifies the name of one or more volumes to expand to the new size.

- Specifies that you want to read the names of one or more volumes to expand from stdin (for example, - << /work/vol\_list.txt).

## Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

The volume can only be expanded. You cannot make a volume smaller.

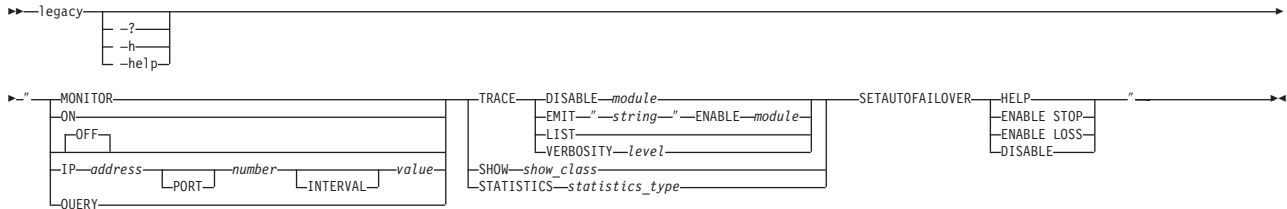
## Example

**Expand a volume** The following example expands the volume myOtherVol:

```
sfscli> expandvol -client superClient myOtherVol
Volume myOtherVol was expanded successfully.
```

# legacy

Runs commands in the legacy command-line interface.



## Parameters

### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### MONITOR

Controls the real time monitor that runs as a part of a server and sends performance statistics to the central monitoring facility for the cluster.

**Note:** This parameter should be used only by trained service technicians.

**ON** Starts the real time monitor.

**OFF** Stops the real time monitor.

**IP address**

Specifies the IP address for the central monitoring facility.

**PORT number**

Specifies the port number for the central monitoring facility. The default is 2700.

**INTERVAL value**

Specifies the time duration in seconds after which the server collects and sends statistics. The default is 20 seconds.

### QUERY

Displays real time monitor status and parameters.

### TRACE

Configures tracing functions. If this parameter is specified without any additional options, this command prints online help about all the trace command variants

#### DISABLE module

Disables tracing on the specified module. If you do not specify a module, this command prints out all of the disabled trace modules. You can use single-wildcard (?) and multiple-wildcard characters (\*) to specify more than one module

**Example:** TM.\*

disables all of the TM modules

**Example:** WAL:WALF????:\*

disables the WAL:WALFLUSH:WALWRITERREP and WAL:WALFORCE:WALWRITERREP modules.

**EMIT** *"string"*

Write the specified string to the trace log,  
/usr/tank/server/log/log.trace.

**ENABLE** *module*

Enables tracing on the specified module. If you do not specify a module, this command prints out all of the enabled trace modules. You can use single-wildcard (?) and multiple-wildcard characters (\*) to specify more than one module (for example, TM:\* would enable all of the TM modules, and WAL:WALF????:\* would enable the WAL:WALFLUSH:WALWRITERREP and WAL:WALFORCE:WALWRITERREP modules.

**Example:** TM.\*

enables all of the TM modules

**Example:** WAL:WALF????:\*

enables the WAL:WALFLUSH:WALWRITERREP and WAL:WALFORCE:WALWRITERREP modules.

**LIST** Displays a list of all the trace modules that you can enable or disable in the metadata server. If a module is enabled, it is prefixed with an asterisk (\*).

**VERBOSITY** *level*

Sets amount of output that is generated from tracing, based on the verbosity level. You can specify a value from 0 to 9. A value of 0, which is the default, emits only the most important messages, whereas a value of 9 emits all messages. If you do not specify a level, the current verbosity level is displayed.

**SHOW** *show\_class*

Displays information about the specified show class.

**Note:** This parameter should be used only by trained service technicians. You can specify one of the following show classes:

**ASMSESSIONS**

Displays information about the administrative sessions.

**BUFPOOL** [*master*]

Displays information about pages in the buffer pool, both the clean list and dirty list. Clean pages are pages available for reuse. The value for *master* is a flag that can be set to 0 or 1. The default is 0.

If the master metadata server is not specified, this command displays the buffer pool of the subordinate metadata server's workload. Specifying SHOW BUFPOOL 1 displays the buffer pool of the master metadata server's workload.

**CLEANLIST** [*master*]

Displays information about only clean pages in the buffer pool (see the **BUFPOOL** keyword).

**CMATTACHPOINTS | CMATT**

Displays the attach-points table.



**CMARENAS**

Displays the arenas table.

**Tip:** The output for this command can be quite large.

**CMCP**

Displays copy partition statistics.

**CMFILESETS | CMCONT**

Displays information about the filesets served by the subordinate metadata server.

**CMLOADMAP** *[detail]* | **CMLOAD** *[detail]*

Displays the load map, which maps the write-ahead log to the metadata server. If the value for *detail* is greater than 0, this keyword displays additional information.

**CMPOLSTAT** *fileset*

Displays storage pool use by a fileset.

**CMSMASTERREC | CMM**

Displays the contents of the Cluster Manager master record.

**CMSECTOR** *sector\_number*

Reads the raw contents of a Cluster Manager sector *sector\_number* from Logical Volume Manager (LVM) and displays the data in dump format.

**CMSPACERECLAIM | CMSPREC**

Displays information about the space-reclamation thread on the master metadata server.

**CMSTGPOLLS | CMSTG**

Displays information about the storage pools.

**CMSVCCLASSES | CMSVC**

Displays the service-class table.

**CMVOLUMES | CMVOL**

Displays the volumes table.

**CMCONTBIND | CMCB**

Displays the fileset-bindings table, which maps the write-ahead-log volume to which each fileset is bound.

**CMWALVOLS | MWALS**

Displays the write-ahead-log volumes table.

**CONDITION | COND**

Displays information about the condition of the metadata server.

**DBPAGE** *space\_ID page\_address* *[format]* *[master]*

Displays the contents of a page.

**DBPAGEHDR** *space\_ID page\_address* *[format]* *[master]*

Displays the contents of a page (header only).

**DBPAGETABLE** *[master]*

Displays information about locked pages (see the **BUFPOOL** keyword).

**DBSPACE** *[master]*

Displays information about DB Spaces (see the **BUFPOOL** keyword).

**DBTXNTABLE** *[master]*

Displays information about in-flight transactions (see the **BUFPOOL** keyword).

**DIODISKS**

Displays the open disk table.

**DIRTYLIST** [*master*]

Displays information about only dirty pages in the buffer pool (see the **BUFPOOL** keyword).

**DISPATCHERSTATS** | **DSPSTATS**

Displays dispatcher and queue statistics.

**FLASHCLEANER** *fileset\_ID* [*flags*] [*dump\_ID*]

Displays information about a fileset's FlashCopy images.

**FLASHTABLE** [*fileset\_ID*]

Displays information about a fileset's FlashCopy image table.

**FSCK** Displays information about an active metadata check that is in progress.

**FSCKSUB**

Displays information about an active metadata check that is in progress on a subordinate metadata server.

**GIODISKS**

Displays information about the global disk.

**GSNODES**

Displays information about the engine.

**INDEXSTATS** *space\_id page\_address* [*format*] [*master*] | **INDEX** *space\_id page\_address* [*format*] [*master*]

Verifies the structural integrity of an index, and shows statistics on the index as a side-effect.

**LATCH**

Displays all active latches.

**LATCHSTATS**

Displays latch contention statistics.

**LATCHX**

Displays all active exclusive latches.

**LATCHXSTATS**

Displays exclusive-latch contention statistics.

**LMCLIENTS**

Displays a list of clients that are known to the lock manager.

**LMALLLOCKS**

Display all of the locks in the lock manager.

**LMLOCKS** *BYCLIENT* [*client\_ID*]

Display all of the locks held by the specified client. To display a list all registered clients, use the **TMCLIENTS** keyword.

**LMLOCKS** *BYOID* [*O*] [*I*] [*D*] | **LMLOCKS** [*O*] [*I*] [*D*]

Display all of the locks held for the specified object.

**LMSTATS**

Displays summary statistics on the lock manager.

**LVMDISKS**

Displays the disk table.

### **LVMMASTERREC | LVMMR**

Displays master control block, including:

- Global ID of master disk.
- Sector size of master disk.
- Size, in bytes, of the logical and physical partitions for this LVM installation.
- Installation common sector size, bytes.
- Number of sectors per partition.
- Number of physical partitions reserved to hold LVM persistent metadata.
- Starting sector numbers for the shadow copies of the LVM persistent tables.
- Index (0 or 1) of the shadow copy that contains the committed copy of the LVM persistent tables.
- Update sequence (version) number.

### **LVMPARTMAP | LVMPM *volume\_ID***

Displays a volume's partition map.

### **LVMREP**

Displays information about the volume manager on a subordinate metadata server.

### **LVMVOLDESC *volume\_ID***

Displays information about a volume's description.

### **LVMVOLUMES**

Displays the volume table, including the following information:

- Logical volume class.
- Logical volume ID.
- Size, in bytes, of logical pages.
- Capacity, in number of logical partitions.
- Number of formatted logical partitions.
- Node ID of subordinate that has locked this logical volume.
- Partition map for each logical partition, including the following data:
  - Logical partition.
  - LVM-assigned disk number of physical disk.
  - Partition status flags.
  - Physical partition number on disk.

### **MASTERWALSTATS | MWALSTATS**

Displays write-ahead-log statistics for the master metadata server.

### **MUTEX**

Displays all active mutexes (mutual exclusion locks).

### **MUTEXSTATS**

Displays statistics about mutexes.

### **OMFILESETS | OMCONT [*fileset\_ID*]**

Displays all fileset information.

### **OMFILESETSTATS | OMCS [*fileset\_ID*]**

Displays statistics about a specific fileset.

### **OMOBJ [*raw* | *pit* | *write* | *main*] [*object*] [*count count*] [*start start*]**

Displays object.

### **OPTCONFIG**

Displays information on the metadata server configurable parameters.

**PMCONFLICTS** [**ALL** | *command*]

Displays a list of conflicting commands for each of all the existing commands or for a specified command.

**PMTABLE**

Displays information about long-running administrative commands.

**SCARENASTATS** | **SCAS** [*fileset\_ID*] [*pool\_ID*]

Displays statistics about a fileset's free-space map as well as the free-space map cache.

**SCSTSDSTATS** | **SCSS** [*fileset\_ID*]

Displays statistics about a fileset's security descriptor table.

**SERVERHANG**

Displays the major data structures used for detecting when a metadata server hangs.

**STATEINFO**

Displays information about the state of the cluster and metadata servers.

**SUBWALSTATS**

Displays write-ahead-log statistics for the subordinate metadata server.

**THREAD** (**STATE** [**force**]) | (**STACK** [**ALL** | *thread\_ID*]) | **USAGE** | **HELP**

Displays information about threads.

**TMATIMESTATS**

Displays information about transaction manager access time attribute (tmatime) statistics.

**TMATIMESTATSRESET**

Resets information about tmatime statistics.

**TMCLIENTS**

Displays statistics for registered clients.

**TMFILTERS**

Displays information about metadata server transaction-manager filters.

**TMPATH** *cluster\_ID fileset\_ID local\_OID* [*epoch\_ID*]

Displays information about metadata server transaction-manager filters.

**VERSION**

Displays information about the metadata server version.

**WALCKPT** [*master*]

Displays write-ahead-log context for the local node (see the **BUFPOOL** keyword).

**WALSTATS** [*master*]

Displays write-ahead-log statistics (see the **BUFPOOL** keyword).

**WALWRITER** [*master*]

Displays write-ahead-log statistics (see the **BUFPOOL** keyword).

**STATISTICS** *statistics\_type*

Displays information about the specified show class. You can specify one of the following types:

**RESET**

Resets the statistics buffer.

**SHOWALL**

Displays all statistics about the Cluster Manager master transactions and the SAN File System protocol received by the master and subordinate metadata servers.

**SHOWBTREE**

Displays B-tree statistics. A B-tree is a file access structure that provides a balanced search tree depth when accessing records. B-trees are used by SAN File System to store file system object and metadata information.

**SHOWBUF**

Displays statistics about the buffer pool.

**SHOWCM**

Displays statistics about the Cluster Manager master transactions received by the master and subordinate metadata server.

**SHOWDSP**

Displays dispatcher and queue statistics.

**SHOWGIO**

Displays statistics about the global disk table.

**SHOWLATCH**

Displays statistics about latches.

**SHOWLATCHS**

Displays statistics about extended latches.

**SHOWLM**

Displays statistics about the lock manager.

**SHOWMUTEX**

Displays statistics about mutexes.

**SHOWSTP**

Displays statistics about the SAN File System protocol by the message type that is received by the metadata servers.

**SHOWWAL**

Displays write-ahead-log statistics for the master and subordinate metadata server.

**SETAUTOFAILOVER**

Enables or disables the failover script.

**HELP** Displays a short help message.

**ENABLE STOP**

Enables the failover script to run for a **stopserver** command.

**ENABLE LOSS**

Enables the failover script to run for a server loss.

**DISABLE**

Disables all failover scripts from running.

**Prerequisites**

This task must be performed only by trained service technicians.

Description

If the legacy command string contains single quotation marks (') or double quotation marks ("), you must precede the character with a backslash (\).

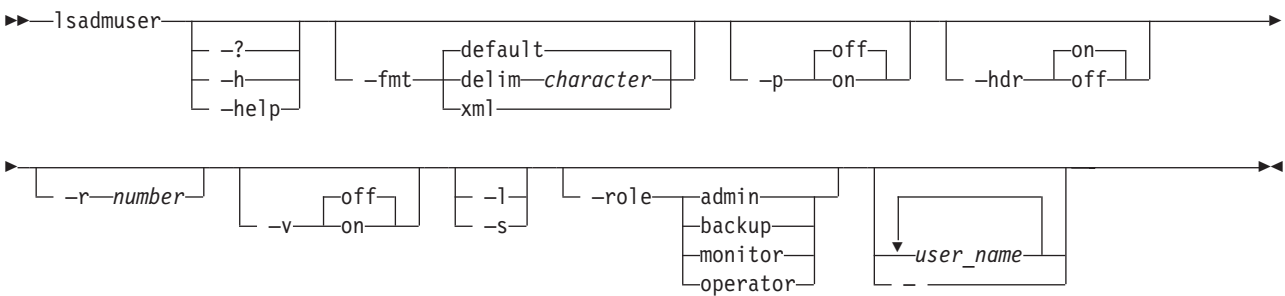
Example

**Running a legacy command** The following example runs the show command from the legacy CLI:

```
sfscli> legacy "SHOW TMCLIENTS"
List of identified clients:
 ClientId=1010513811, ClientName="st-client-1", SendInProg=0, SendMsgNo=2,
 ccb_sendAcked=1, DeliveredMsgNo=1, IPaddress=192.168.1.10, IPport= 32769
```

lsadmuser

Displays a list of administrative users and their attributes.



Parameters

**-? | -h | -help**  
Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

**-fmt**  
Specifies the format of the output. You can specify one of the following values:

**default**  
Specifies to display output in a tabular format using spaces as the delimiter between the columns. This is the default value. For example:

Name	User	Role	Authorization	Authorization Timeout (secs)
maki	Admin	Current		300

**delim character**  
Specifies to display output in a tabular format using the specified character to separate the columns. If you use a shell metacharacter (for example, \* or \t) as the delimiting character, enclose the character in single quotation marks (') or double quotation marks ("). A blank space is not a valid character. For example:  
maki,Admin,Current,300 secs

**xml** Specifies to display output using XML format, for example:  
<IRETURNVALUE>  
<INSTANCE CLASSNAME="STC\_AdminUser">  
<PROPERTY NAME="Name" TYPE="string"><VALUE TYPE='string'>maki</VALUE>  
</PROPERTY>

```

<PROPERTY NAME="EffectiveRole" TYPE="unit16"><VALUE="unit16">3</VALUE>
</PROPERTY>
<PROPERTY NAME="IsAuthorizationCurrent" TYPE="boolean"><VALUE
TYPE="boolean">true</VALUE></PROPERTY>
<PROPERTY NAME="AuthCurrentRemainingTime" TYPE="uint32"><VALUE
TYPE="uint32">300</VALUE></PROPERTY>
</INSTANCE>
</IRETURNVALUE>

```

**-p** Specifies whether to display one page of text at a time or all text at once.

**off** Displays all text at one time. This is the default value when the **sfscli** command is run in single-shot mode.

**on** Displays one page of text at time. Pressing any key displays the next page. This is the default value when the **sfscli** command is run in interactive mode.

**-hdr**

Specifies whether to display the table header.

**on** Displays the table header. This is the default value.

**off** Does not display the table header.

**-r number**

Specifies the number of rows per page to display when the **-p** parameter is on. The default is 24 rows. You can specify a value from 1 to 100.

**-v** Specifies whether to enable verbose mode.

**off** Disables verbose mode. This is the default value.

**on** Enables verbose mode.

**-l** Displays additional information about the administrative users.

**-s** Displays only the names of the administrative users.

**-role**

Displays information for only those users with the specified user role . Possible roles are:

**admin** Shows only those users assigned to the Administrator role.

**backup**

Shows only those users assigned to the Backup role.

**monitor**

Shows only those users assigned to the Monitor role.

**operator**

Shows only those users assigned to the Operator role.

*user\_name*

Specifies the names of one or more administrative users to list. If not specified, this command lists all of the administrative users assigned to all user roles.

– Specifies that you want to read the names of the one or more administrative users to list from stdin (for example, – << /work/users\_list.txt).

## Description

**Note:** This command operates only against the local engine.

Administrative users are defined in the LDAP server as authorized to perform administrative actions on the SAN File System. A user's role is determined by its membership in an LDAP group that is associated with the various SAN File System roles.

If you do not specify a listing parameter, this command displays the following default information for each administrative user:

- Name of the administrative user.
- Role assigned to the administrative user.
- Authorization (Current or Not Current).

If you specify the `-l` listing parameter, this command displays the following information in addition to the default information for each administrative user:

- Authorization timeout, in seconds.

**Example**

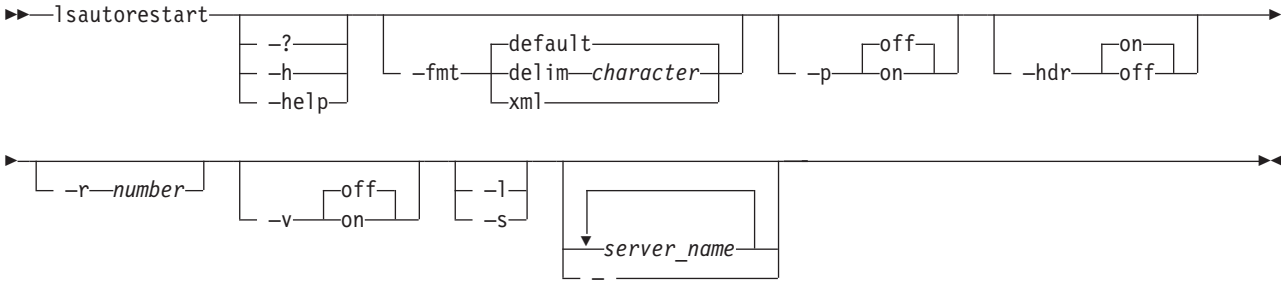
**Lists all administrative users** The following example lists all SAN File System administrative users and additional information about their assigned roles:

```
sfscli> lsadmuser -l
```

Name	User Role	Authorization	Authorization Timeout (secs)
maki	Admin	Current	300
saki	Backup	Not Current	0
mizi	Operator	Current	465
jeko	Monitor	Not Current	0

**Isautorestart**

Displays a list of metadata servers and the automatic-restart settings for each.



**Parameters**

`-? | -h | -help`

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

`-fmt`

Specifies the format of the output. You can specify one of the following values:

**default**

Specify this value to display output in a tabular format using spaces as the delimiter between the columns. This is the default value. For example:



Name	Service State	Last Probe State	Last Probe	Probes
ST1	Running	Live Server	Feb 11, 2003 9:35:47 AM	218690

Highest Retries

2

**delim** *character*

Specify this value to display output in a tabular format using the specified character to separate the columns. If you use a shell metacharacter (for example, \* or \t) as the delimiting character, enclose the character in single quotation marks (') or double quotation marks ("). A blank space is not a valid character. For example:

ST1,Running,Live Server,Feb 11, 2003 9:35:47 AM,218690,2

**xml** Specify this value to display output using XML format, for example:

```
<IRETURNVALUE>
<INSTANCE CLASSNAME="STC_TankWatchdog">
<PROPERTY NAME="ServiceName" TYPE="string"><VALUE TYPE="string">ST1
</VALUE></PROPERTY>
<PROPERTY NAME="State" TYPE="uint32"><VALUE TYPE="uint32">1</VALUE>
</PROPERTY>
<PROPERTY NAME="ProbeState" TYPE="uint32"><VALUE TYPE="uint32">2</VALUE>
</PROPERTY>
<PROPERTY NAME="LastProbeTimeStamp" TYPE="datetime"><VALUE
TYPE="datetime">20030211093547.000009-420</VALUE></PROPERTY>
<PROPERTY NAME="TotalProbes" TYPE="uint64"><VALUE TYPE="uint64">218690
</VALUE></PROPERTY>
<PROPERTY NAME="RetriesHWM" TYPE="uint32"><VALUE TYPE="uint32">2</VALUE>
</PROPERTY>
</INSTANCE>
</IRETURNVALUE>
```

**-p** Specifies whether to display one page of text at a time or all text at once.

**off** Displays all text at one time. This is the default value when the sfscli command is run in single-shot mode.

**on** Displays one page of text at time. Pressing any key displays the next page. This is the default value when the sfscli command is run in interactive mode.

**-hdr**

Specifies whether to display the table header.

**on** Displays the table header. This is the default value.

**off** Does not display the table header.

**-r** *number*

Specifies the number of rows per page to display when the **-p** parameter is on. The default is 24 rows. You can specify a value from 1 to 100.

**-v** Specifies whether to enable verbose mode.

**off** Disables verbose mode. This is the default value.

**on** Enables verbose mode.

**-l** Displays additional information about the metadata servers.

**-s** Displays default information about the metadata servers.

*server\_name*

Specifies the names of one or more metadata servers to display. If not specified, settings for all metadata servers are displayed.

- Specifies that you want to read the names of one or more metadata servers to display from stdin (for example, – << /work/server\_list.txt).

## Description

If you run this command from an engine hosting a subordinate metadata server, you can display information about only the local metadata server. If you run this command from the engine hosting the master metadata server, you can display information about any metadata server.

If you do not specify a listing parameter, this command displays the following default information for each metadata server:

- Name of the metadata server in the cluster.
- State of the metadata server restart service. Possible states are:
  - Off** The metadata server restart service is not running.
  - Failed** The metadata server restart service is running but has failed to restart the metadata server after several attempts.
  - Running** The metadata server restart service is running and ready to restart the metadata server if it fails.
  - Standby** The metadata server restart service is running, but the metadata server has been manually shut down. You must manually restart metadata server to return the metadata server restart service to the running state.
  - Unknown** The metadata server restart service is in an unknown state because the metadata server could not be reached.
- Probed metadata server status. Possible status values are:
  - Absent Server** The metadata server restart service could not find the metadata server and attempts to restart the metadata server.
  - Live Server** The metadata server restart service found the metadata server and it is up and running. There is no need to restart the metadata server.
  - Not Probed** The metadata server restart service has not started probing the metadata server because it is either in the off or standby state.
  - Probing** The metadata server restart service has started probing the metadata servers for status.
  - Unknown** The metadata server restart service cannot determine whether the metadata server is live or absent.
 

It is possible for the probe state to be unknown when the metadata server is live.

Example: If the metadata server fails, the metadata server restart service determines that the metadata server is absent, and restarts it. If the metadata server is slow to start, on the next probe cycle, the metadata server is neither live nor absent. When it is probed again, the metadata server is live.

Example: If the local disk is full, and the metadata server restart service tries to restart the metadata server but cannot write to its log file. The

metadata server crashes, and the metadata server restart service tries to restart it again. On the next probe cycle, the state of the metadata server is absent or unknown.

- Timestamp when the metadata server was last probed for status.
- Total number of probes that the metadata server restart service performed.
- Highest number of times that the metadata server restart service has attempted to restart the metadata server.

If you specify the `-l` listing parameter, this command displays the following information in addition to the default information for each metadata server:

- Timestamp when the automatic-restart service started on the metadata server.
- Amount of time, in seconds, between each probe.
- Current number of times that the automatic-restart service has attempted to restart the metadata server.
- Total number of times that the automatic-restart service has attempted to restart the metadata server.
- Maximum number of times that the automatic-restart service attempts to restart the metadata server.
- Lowest number of times that the automatic-restart service has attempted to restart the metadata server.
- Total number of times that a probe for a live metadata server has taken longer than the test timeout interval.
- Maximum amount of time, in milliseconds, to wait for a metadata server to respond before declaring it down.
- Amount of time, in seconds, used the last time the metadata server was probed to see if it is running.
- Highest amount of time, in milliseconds, taken to determine if the metadata server is running.
- Lowest amount of time, in milliseconds, taken to determine if the metadata server is running.
- Number of times the watchdog probed the metadata server to determine if it is absent.
- Amount of time, in milliseconds, used the last time the metadata server was probed to determine if it is absent.
- Highest amount of time, in milliseconds, taken to determine if the metadata server is absent.
- Lowest amount of time, in milliseconds, taken to determine if the metadata server is absent.
- Amount of time to wait for the metadata server to restart before the metadata server is declared dead. If the metadata server restart timeout or the retry limit is exceeded, the automatic restart service is disabled.

### Example

**Displays the automatic-restart settings** The following example displays the automatic-restart settings and additional information for all metadata servers in the cluster:

```
sfscli> lsautorestart ST1
```

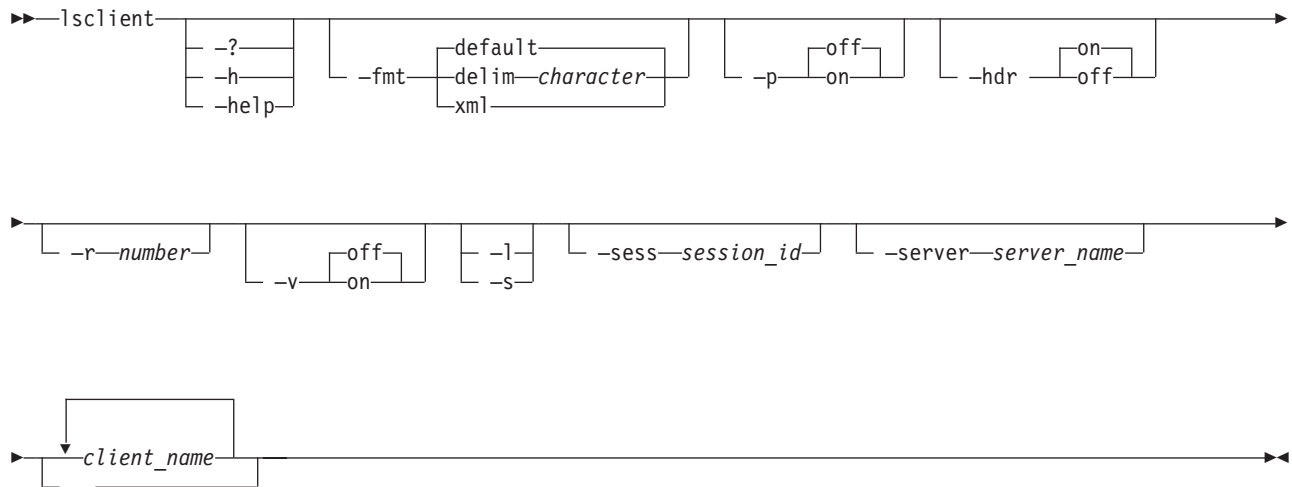
Name	Service State	Last Probe State	Last Probe	Probes
ST1	Running	Live Server	Feb 11, 2003 9:35:47 AM	5

Highest Retries

```
=====
2
```

## Isclient

Displays a list of clients that are currently being served by one or more metadata servers in the cluster.



### Parameters

#### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### -fmt

Specifies the format of the output. You can specify one of the following values:

##### default

Specify this value to display output in a tabular format using spaces as the delimiter between the columns. This is the default value. For example:

Client	Session ID	State	Server	Renewals	Privilege
jeko	1	Current	ST1	45	Standard

##### delim character

Specify this value to display output in a tabular format using the specified character to separate the columns. If you use a shell metacharacter (for example, \* or \t) as the delimiting character, enclose the character in single quotation marks (') or double quotation marks ("). A blank space is not a valid character. For example:

```
jeko,1,Current,ST1,45,Standard
```

##### xml

Specify this value to display output using XML format, for example:

```
<IRETURNVALUE>
<INSTANCE CLASSNAME="STC_RegisteredFSClient">
 <PROPERTY NAME="Name" TYPE="string"><VALUE TYPE="string">jeko</VALUE>
</PROPERTY>
 <PROPERTY NAME="Id" TYPE="unit64"><VALUE TYPE="unit64">1</VALUE>
</PROPERTY>
 <PROPERTY NAME="State" TYPE="unit16"><VALUE TYPE="unit16">1</VALUE>
</PROPERTY>
 <PROPERTY NAME="SystemName" TYPE="string"><VALUE TYPE="string">ST1
</VALUE></PROPERTY>
 <PROPERTY NAME="LeaseRenewals" TYPE="uint64"><VALUE TYPE="uint64">45
```

```

</VALUE></PROPERTY>
<PROPERTY NAME="IsPrivilege" TYPE="boolean"><VALUE TYPE="boolean">true
</VALUE></PROPERTY>
</INSTANCE>
</IRETURNVALUE>

```

- p** Specifies whether to display one page of text at a time or all text at once.
  - off** Displays all text at one time. This is the default value when the **sfscli** command is run in single-shot mode.
  - on** Displays one page of text at time. Pressing any key displays the next page. This is the default value when the **sfscli** command is run in interactive mode.
- hdr**
  - Specifies whether to display the table header.
  - on** Displays the table header. This is the default value.
  - off** Does not display the table header.
- r** *number*
  - Specifies the number of rows per page to display when the **-p** parameter is on. The default is 24 rows. You can specify a value from 1 to 100.
- v** Specifies whether to enable verbose mode.
  - off** Disables verbose mode. This is the default value.
  - on** Enables verbose mode.
- l** Displays additional information about each client.
- s** Displays default information about each client.
- sess** *session\_id*
  - Specifies the client-session ID to list. If not specified, this command displays a list of all clients.
- server** *server\_name*
  - Specifies the name of a metadata server to query for clients. If specified, this command displays only those clients that are served by the specified metadata server. If not specified, this command displays clients that are served by all metadata servers in the cluster.
- client\_name*
  - Specifies the names of one or more clients to list. If not specified, this command displays all clients that are being served by the specified metadata server.
- Specifies that you want to read the names of one or more clients to list from stdin (for example, **- << /work/client\_list.txt**).

## Description

If you run this command from an engine hosting a subordinate metadata server, you can display information about only the local metadata server. If you run this command from the engine hosting the master metadata server, client-server connections for the entire cluster are displayed.

**Note:** This command does not display a server as serving a client until the client accesses one of the filesets assigned to that server.

When you specify multiple parameters in this command, these behaviors apply:

- Flags work as AND coordinators.
- Objects work as OR coordinators.
- When flags and objects are combined, they are listed in this manner:  
WHERE *flag\_1* AND *flag\_2* AND ... AND *flag\_n* AND (*object\_1* OR *object\_2* OR ... OR *object\_m*)
- When flags or objects are not required and not specified, this command searches all possible values unless a default overrides this rule.

For example, if you specify **-sess**, **-server**, and two client names, the search would be performed in the following manner:

WHERE *session\_ID* AND *server\_name* AND (*client\_name\_1* OR *client\_name\_2*)

If you do not specify a listing parameter, this command displays the following default information for each client:

- Client name. This name is configured by the administrator on the client node.
- Client-session ID. Each client can have one session running on each metadata server simultaneously.
- State of the client. Possible states are:

**Current**

The client is currently being served by one of the metadata servers.

**Expired**

The client is not currently being served by one of the metadata servers.

- Metadata server that is serving the client.
- Number of renewals since the client-session was started.
- Client privilege or access level. Possible privileges are:

**Root** Indicates that the user has root privileges, and is considered a privileged user.

**Standard**

Indicates that the user has default access level.

If you specify the **-s** listing parameter, this command displays only the following information for each client:

- Client name.
- Client-session ID.
- Metadata server that is serving the client.

If you specify the **-l** listing parameter, this command displays the following information in addition to the default information for each client:

- Date and time of the last client-session renewal.
- Time until the next client-session renewal is to occur, in seconds.
- Client IP address.
- Client port number.
- Client operating-system platform.
- Client file-system driver version.
- Number of transactions started, per client session.
- Number of transactions that have completed, per client session.
- Number of session locks, per client session.

- Number of data locks, per client session.
- Number of byte-range locks, per client session.

**Tip:** The metadata server does not know which clients are connected to SAN File System. If you specify a client that is not connected, this command will return a warning that no rows were found.

### Example

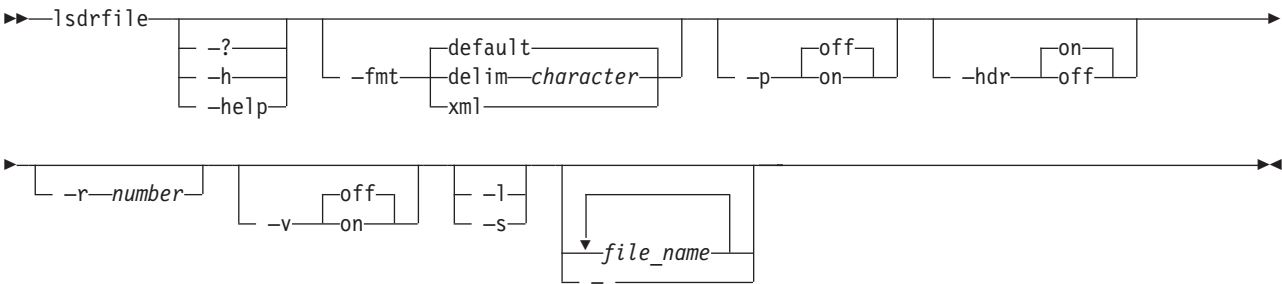
**List clients** The following example displays information about all clients currently being served by metadata server ST1:

```
sfscli> lsclient -server ST0,ST1
```

Client	Session ID	State	Server	Renewals	Privilege
client01	ABCD1234FBXC5678	Current	ST1	45	Standard
client02	ABCD1234FBXC5679	Current	ST1	1546	Root
client03	ABCD1234FBXC5670	Expired	ST0	587	Standard

## lsdrfile

Displays a list of system-metadata disaster-recovery files.



### Parameters

- ? | -h | -help**  
Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.
- fmt**  
Specifies the format of the output. You can specify one of the following values:
  - default**  
Specifies to display output in a tabular format using spaces as the delimiter between the columns. This is the default value. For example:  

```
Name Date and Time Size (KB)
=====
SysMetadata Jan 3, 2003 4:35:46 PM 10000
```
  - delim character**  
Specifies to display output in a tabular format using the specified character to separate the columns. If you use a shell metacharacter (for example, \* or \t) as the delimiting character, enclose the character in single quotation marks (') or double quotation marks ("). A blank space is not a valid character. For example:  

```
SysMetadata,Jan 3, 2003 4:35:46 PM,10000 KB
```

**xml** Specifies to display output using XML format, for example:

```
<IRETURNVALUE>
<INSTANCE CLASSNAME="STC_SystemMDRAid">
<PROPERTY NAME="Name" TYPE="string"><VALUE TYPE="string">SysMetadata
</VALUE></PROPERTY>
<PROPERTY NAME="InstallDate" TYPE="datetime"><VALUE TYPE="datetime">
20030103043546.000000&45;420</VALUE></PROPERTY>
<PROPERTY NAME="Size" TYPE="uint64"><VALUE TYPE="uint64">10000</VALUE>
</PROPERTY>
h</INSTANCE>
</IRETURNVALUE>
```

**-p** Specifies whether to display one page of text at a time or all text at once.

**off** Displays all text at one time. This is the default value when the **sfscli** command is run in single-shot mode.

**on** Displays one page of text at time. Pressing any key displays the next page. This is the default value when the **sfscli** command is run in interactive mode.

**-hdr**

Specifies whether to display the table header.

**on** Displays the table header. This is the default value.

**off** Does not display the table header.

**-r number**

Specifies the number of rows per page to display when the **-p** parameter is on. The default is 24 rows. You can specify a value from 1 to 100.

**-v** Specifies whether to enable verbose mode.

**off** Disables verbose mode. This is the default value.

**on** Enables verbose mode.

**-l** Displays the default information for the system-metadata disaster-recovery files.

**-s** Displays only the names of the system-metadata disaster-recovery files.

*file\_name*

Specifies the name of one or more system-metadata disaster-recovery files to display. If not specified, this command displays information for all system-metadata disaster-recovery files.

**-** Specifies the file from which this commands reads the names of one or more system-metadata disaster-recovery files to display.

## Description

**Note:** This command operates only against the local engine.

If you do not specify a listing parameter or if you specify the **-l** listing parameter, this command displays the following default information for each system-metadata disaster-recovery file:

- Name of the disaster-recovery file.
- Date and time the disaster-recovery file was created.
- Size of the disaster-recovery file, in KB.



Example

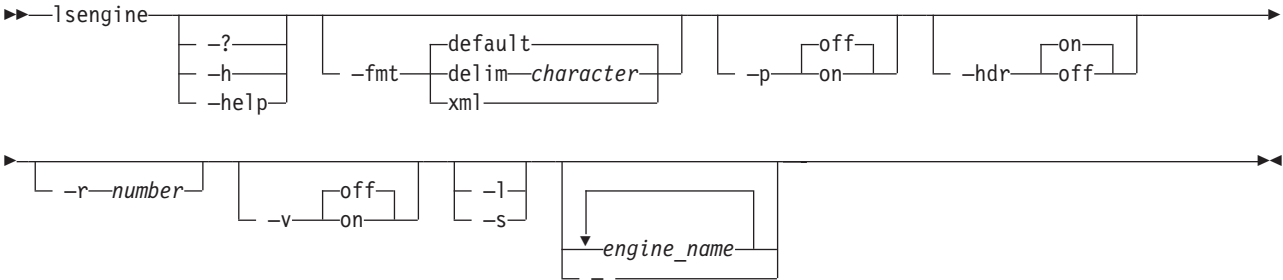
**List all disaster-recovery files** The following example displays information for all disaster-recovery files:

```
sfscli> lsdrfile

Name Date and Time Size (KB)
=====
SysMetadata Jan 3, 2003 4:35:46 PM 10000
```

lsengine

Displays a list of storage engines and their attributes.



Parameters

**-? | -h | -help**  
Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

**-fmt**  
Specifies the format of the output. You can specify one of the following values:

**default**  
Specify this value to display output in a tabular format using spaces as the delimiter between the columns. This is the default value. For example:

Name	Engine IP	Boot State	Temp	Fans	Voltage
ST0	129.42.16.99	In POST	Warning	Normal	Normal

**delim character**  
Specify this value to display output in a tabular format using the specified character to separate the columns. If you use a shell metacharacter (for example, \* or \t) as the delimiting character, enclose the character in single quotation marks (') or double quotation marks ("). A blank space is not a valid character. For example:  
ST0,129.42.16.99,In POST,Warning,Normal,Normal

**xml** Specify this value to display output using XML format, for example:  
<IRETURNVALUE>  
<INSTANCE CLASSNAME="STC\_ComputerSystem">  
<PROPERTY NAME="CurrentState" TYPE="uint32"><VALUE TYPE="uint32">1  
</VALUE></PROPERTY>  
<PROPERTY NAME="Name" TYPE="string"><VALUE TYPE="string">ST0</VALUE>  
</PROPERTY>  
</INSTANCE>  
</IRETURNVALUE>

- p** Specifies whether to display one page of text at a time or all text at once.
  - off** Displays all text at one time. This is the default value when the **sfscli** command is run in single-shot mode.
  - on** Displays one page of text at time. Pressing any key displays the next page. This is the default value when the **sfscli** command is run in interactive mode.
- hdr**
  - Specifies whether to display the table header.
  - on** Displays the table header. This is the default value.
  - off** Does not display the table header.
- r *number***
  - Specifies the number of rows per page to display when the **-p** parameter is on. The default is 24 rows. You can specify a value from 1 to 100.
- v** Specifies whether to enable verbose mode.
  - off** Disables verbose mode. This is the default value.
  - on** Enables verbose mode.
- l** Displays the default information for each engine. This is the default.
- s** Displays only the IP address and name of each engine.
- engine\_name***
  - Specifies the names of one or more engines to display. If not specified, this command displays information for all engines.
- Specifies the file from which this command reads the names of one or more engines to display.

## Description

When you issue this command on a three-node cluster, command processing can take up to three minutes to complete.

If you specify the **-s** listing parameter, this command displays the IP address and name of each engine.

If you do not specify a listing parameter or if you specify the **-l** listing parameter, this command displays the following information for each engine:

- Engine IP address.
- Engine name.
- Boot state. Possible values are:

### Before POST

The engine is powered on but has not started the power-on self test (POST).

### Booted Flash

The engine has read the System BIOS but has not started loading the operating system.

### Booting OS

The engine has stated but not completed loading the operating system.

### CPUs Held in Reset

The engine has been reset after a hardware fault.

**In POST**

The engine is running the POST.

**In OS** The engine is running in the normal state.

**Stopped in POST (error detected)**

The engine is powered on but has not completed the POST due to an error.

**Unknown / Power Off**

The boot state is unknown. The engine could be powered off.

- Temperature state. Possible values are:

- The temperature thresholds are not set.

**Normal**

The temperatures of all components in all engines are below the Warning threshold.

**Warning**

The temperature of one or more engine components are above the Warning threshold.

**Error** The temperature of one or more engine components is higher than the Soft Shutdown threshold.

**Unknown**

The RSA card could not be accessed.

- Fan state. Possible values are:

**Normal**

All fans are operating above 15% of its fan-speed capacity.

**Warning**

One or more fans are operating below 15% of its fan-speed capacity.

**Error** The RSA card could not be accessed.

- Voltage state. Possible values are:

- The voltage thresholds are not set.

**Normal**

The voltages of all components in all engines are above the Warning Low Voltage threshold and below the Warning High Voltage threshold.

**Warning**

The voltage of one or more engine components is below the Warning Low Voltage threshold or above the Warning High Voltage threshold.

**Error** The RSA card could not be accessed.

**Example**

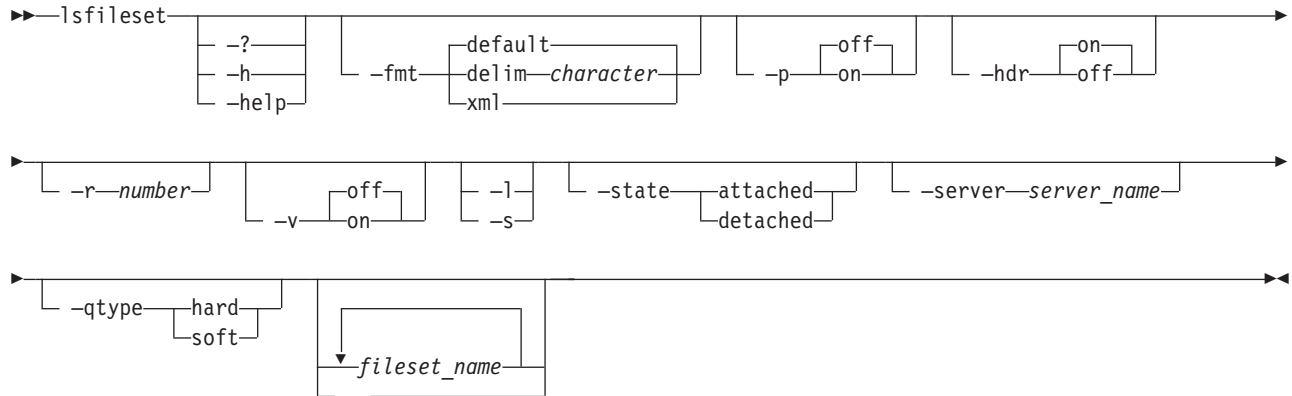
**List engines** The following example displays information about two engines: ST0 and ST1:

```
sfsccli> lsengine ST1 ST2
```

Name	Engine IP	Boot State	Temp	Fans	Voltage
ST0	129.42.16.99	In POST	Warning	Normal	Normal
ST1	129.42.16.98	In OS	Normal	Normal	Normal

# lsfileset

Displays a list of filesets and their attributes.



## Parameters

### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### -fmt

Specifies the format of the output. You can specify one of the following values:

#### default

Specify this value to display output in a tabular format using spaces as the delimiter between the columns. This is the default value. For example:

Name	Fileset	State	Quota Type	Quota (MB)	Used (MB)	Used (%)
cnt_B	Attached	Soft		20	10	50
Threshold (%)    Most Recent Image						
19	Feb 6, 2003	14:54:15 PM				

#### delim *character*

Specify this value to display output in a tabular format using the specified character to separate the columns. If you use a shell metacharacter (for example, \* or \t) as the delimiting character, enclose the character in single quotation marks (') or double quotation marks ("). A blank space is not a valid character. For example:

```
cnt_B,Attached,Soft,20 MB,10 MB,50%,19%,Feb 6, 2003 14:54:15 PM
```

#### xml

Specify this value to display output using XML format, for example:

```
<IRETURNVALUE>
<INSTANCE CLASSNAME="STC_Fileset">
 <PROPERTY NAME="Name" TYPE="string"><VALUE TYPE="string">cnt_B</VALUE>
</PROPERTY>
 <PROPERTY NAME="State" TYPE="uint32"><VALUE TYPE="uint32">1</VALUE>
</PROPERTY>
 <PROPERTY NAME="Quota" TYPE="uint64"><VALUE TYPE="uint64">20</VALUE>
</PROPERTY>
 <PROPERTY NAME="IsHardQuota" TYPE="boolean"><VALUE TYPE="boolean">>false
```

```

</VALUE></PROPERTY>
<PROPERTY NAME="AlertPercentage" TYPE="uint16"><VALUE TYPE="uint16">19
</VALUE></PROPERTY>
<PROPERTY NAME="SizeAllocated" TYPE="uint64"><VALUE TYPE="uint64">10
</VALUE></PROPERTY>
<PROPERTY NAME="SizeAllocatedPercentage" TYPE="uint16"><VALUE
TYPE="uint16">50</VALUE></PROPERTY>
<PROPERTY NAME="LastPITCopyDate" TYPE="datetime"><VALUE TYPE="datetime">
20030206145415.000000+000</VALUE></PROPERTY>
</INSTANCE>
</IRETURNVALUE>

```

**-p** Specifies whether to display one page of text at a time or all text at once.

- off** Displays all text at one time. This is the default value when the **sfscli** command is run in single-shot mode.
- on** Displays one page of text at time. Pressing any key displays the next page. This is the default value when the **sfscli** command is run in interactive mode.

**-hdr**

Specifies whether to display the table header.

- on** Displays the table header. This is the default value.
- off** Does not display the table header.

**-r** *number*

Specifies the number of rows per page to display when the **-p** parameter is on. The default is 24 rows. You can specify a value from 1 to 100.

**-v** Specifies whether to enable verbose mode.

- off** Disables verbose mode. This is the default value.
- on** Enables verbose mode.

**-l** Displays additional information about the filesets.

**-s** Displays only the names of the filesets.

**-state**

Lists only the filesets that match the specified state. Possible states are:

**attached**

Lists only the filesets that are attached to the global namespace.

**detached**

Lists only the filesets that are not attached to the global namespace.

**-server** *server\_name*

Lists only filesets assigned to the specified metadata server.

**-qtype**

Lists only the filesets that match the specified quota type. Possible types are:

- hard** Lists only the filesets that have hard quotas.
- soft** Lists only the filesets that have soft quotas.

*fileset\_name*

Specifies the names of one or more filesets to list. If not specified, this command displays a list of all filesets attached to the global namespace.

- Specifies that you want to read the names of one or more filesets to list from stdin (for example, **- << /work/cnt\_list.txt**).

## Prerequisites

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

When you specify multiple parameters in this command, these behaviors apply:

- Flags work as AND coordinators.
- Objects work as OR coordinators.
- When flags and objects are combined, they are listed in this manner:  
`WHERE flag_1 AND flag_2 AND ... AND flag_n AND (object_1 OR object_2 OR ... OR object_m)`
- When flags or objects are not required and not specified, this command searches all possible values unless a default overrides this rule.

For example, if you specify **-state**, **-qtype**, and two fileset names, the search would be performed in the following manner:

`WHERE state AND quota_type AND (fileset_name_1 OR fileset_name_2)`

If you do not specify a listing parameter, this command displays the following default information for each fileset:

- Name of the fileset.
- State of the fileset (Detached, Idle, or Inuse).
- Quota type.
- Quota size, in MB.
- Amount of space used, in MB.
- Threshold, as a percentage of space used.
- Timestamp of the most recent FlashCopy image.
- Metadata server that hosts the fileset.

**Tip:** If the quota is not set, the values for the amount of space that is used (Used %) and the threshold (Threshold %) are displayed as 0 and have no meaning. Because the threshold value is based on the amount of space used, if the amount of space used is not defined, any value specified for the threshold value has no meaning and no alert is sent if the displayed value is exceeded.

If you specify the **-l** listing parameter, this command displays the following information in addition to the default information for each fileset:

- Assigned server. (A dash (-) indicates that the fileset was dynamically assigned.)
- State of the metadata server.
- Number of FlashCopy images that exist for this fileset.
- Attach point.
- Directory name.
- Directory path.
- Parent fileset.
- Number of nested filesets (children).
- Fileset description.

## Example

**Listing filesets** The following example lists all the available information about all filesets in the global namespace:

sfsccli> **lsfileset**

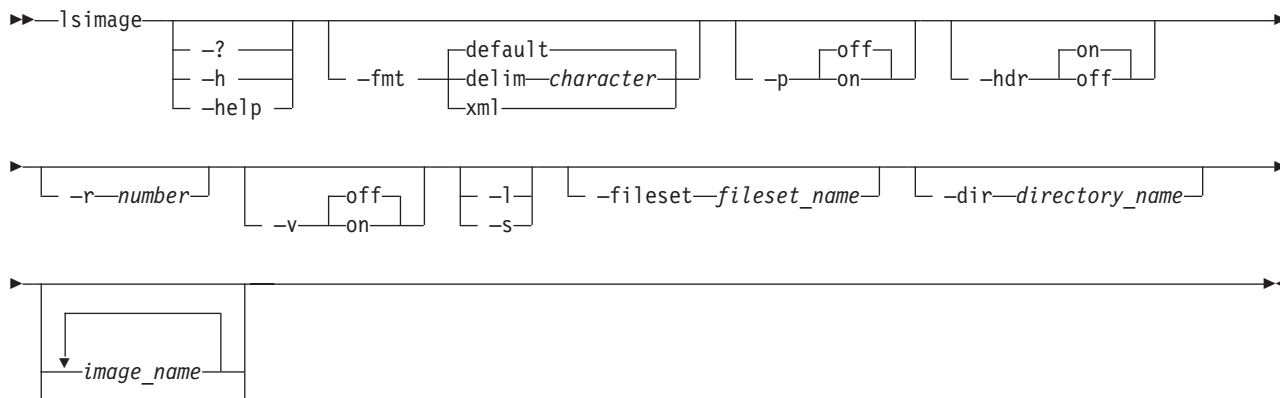
Name	State	Quota (MB) Type	Quota	Used (MB)	Used (%)
ROOT	Attached	Soft	20	10	50
cnt_B	Detached	Hard	88	100	48
cnt_C	Attached	Hard	120	400	45

Threshold (%)	Most Recent Image	Server
19	-	ST0
70	Nov 6, 2003 20:01:48 PM	ST1
2	Mar 7, 2003 10:36:37 PM	ST2

## lsimage

Displays a list of FlashCopy images in all filesets or a specified fileset.



### Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

**-fmt**

Specifies the format of the output. You can specify one of the following values:

**default**

Specifies to display output in a tabular format using spaces as the delimiter between the columns. This is the default value. For example:

Name	Fileset	Directory Name	Date
FCA_image	cnt_A	personnel	Dec 3, 2003 5:48:46 PM

**delim character**

Specifies to display output in a tabular format using the specified character to separate the columns. If you use a shell metacharacter (for example, \* or \t) as the delimiting character, enclose the character in single quotation marks (') or double quotation marks ("). A blank space is not a valid character. For example:

FCA\_image,cnt\_A,personnel,Dec 3, 2003 5:48:46 PM

**xml**

Specifies to display output using XML format, for example:

```

<IRETURNVALUE>
<INSTANCE CLASSNAME="STC_PitImage">
<PROPERTY NAME="FilesetName" TYPE="string"><VALUE TYPE="string">cnt_A<
/VALUE></PROPERTY>
<PROPERTY NAME="Name" TYPE="string"><VALUE TYPE="string">FCA_imag<
/VALUE></PROPERTY>
<PROPERTY NAME="InstallDate" TYPE="datetime"><VALUE TYPE="datetime">
20031203174846.000009-420</VALUE></PROPERTY>
<PROPERTY NAME="DirectoryName" TYPE="string"><VALUE TYPE="string">
personnel</VALUE></PROPERTY>
</INSTANCE>
</IRETURNVALUE>

```

**-p** Specifies whether to display one page of text at a time or all text at once.

**off** Displays all text at one time. This is the default value when the **sfscli** command is run in single-shot mode.

**on** Displays one page of text at time. Pressing any key displays the next page. This is the default value when the **sfscli** command is run in interactive mode.

**-hdr**

Specifies whether to display the table header.

**on** Displays the table header. This is the default value.

**off** Does not display the table header.

**-r** *number*

Specifies the number of rows per page to display when the **-p** parameter is on. The default is 24 rows. You can specify a value from 1 to 100.

**-v** Specifies whether to enable verbose mode.

**off** Disables verbose mode. This is the default value.

**on** Enables verbose mode.

**-l** Displays additional information about the FlashCopy images.

**-s** Displays only the name of the FlashCopy images and the fileset associated with each.

**-fileset** *fileset\_name*

Specifies the name of the fileset for which to list all FlashCopy images. If not specified, this command lists FlashCopy images in all filesets.

**-dir** *directory\_name*

Specifies the directory under the .flashcopy directory from which to list the FlashCopy image. This directory can be up to 256 characters in length and must not contain backslash (\) or forward slash (/) characters as delimiters. If not specified, this command lists the FlashCopy images in all directories under the .flashcopy directory.

**image\_name**

Specifies the names of one or more FlashCopy images to display. If not specified, this command lists all FlashCopy images in the specified fileset and directory.

**-** Specifies that you want to read the names of one or more FlashCopy images to display from stdin (for example, **- << /work/image\_list.txt**).



## Prerequisites

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

When you specify multiple parameters in this command, these behaviors apply:

- Flags work as AND coordinators.
- Objects work as OR coordinators.
- When flags and objects are combined, they are listed in this manner:  
WHERE *flag\_1* AND *flag\_2* AND ... AND *flag\_n* AND (*object\_1* OR *object\_2* OR ... OR *object\_m*).
- When flags or objects are not required and not specified, this command searches all possible values unless a default overrides this rule.

For example, if you specify `-fileset`, `-dir`, and two FlashCopy image names, the search would be performed in the following manner: WHERE `fileset` AND `directory` AND (`image_name_1` OR `image_name_2`).

The most-recently created FlashCopy image are listed first.

If no parameters are specified, this command lists all FlashCopy images in all filesets and directories.

If you do not specify a listing parameter, this command displays the following default information for each FlashCopy image:

- FlashCopy image name.
- FlashCopy image state. If the FlashCopy image is in a complete state, it is intact and you can revert the fileset to that Flashcopy image. If the FlashCopy image is in an incomplete state, it contains incomplete files caused by a (forced) volume deletion. You cannot revert a fileset to that Flashcopy image.
- Fileset name.
- Directory name.
- Timestamp when the FlashCopy image was created.

If you specify the `-l` listing parameter, this command displays the following information in addition to the default information for each FlashCopy image:

- Description.

## Example

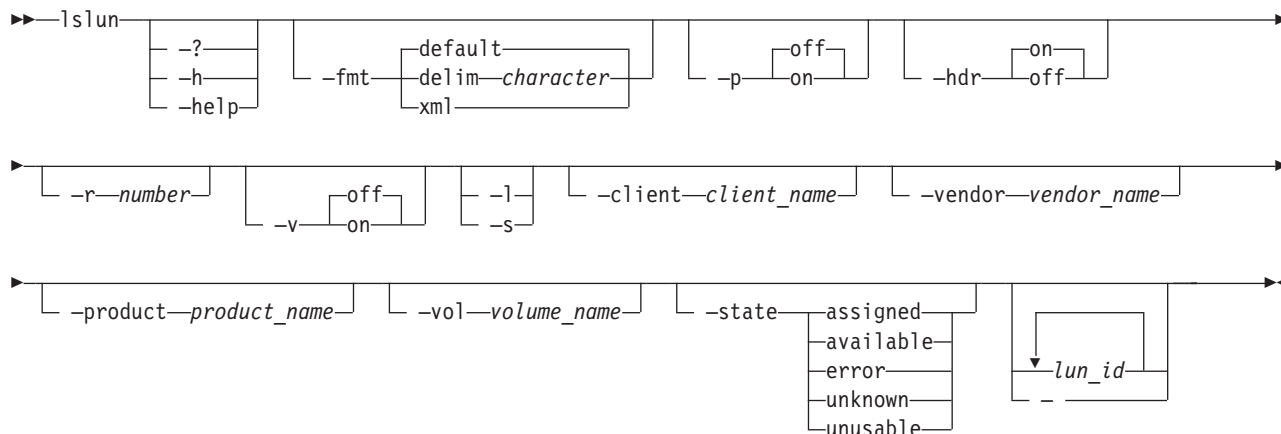
**Listing FlashCopy images** The following example lists all the available information about the FlashCopy images:

```
sfsccli> lsimage -l
```

Name	State	Fileset	Directory Name	Date	Description
FCC_image	Complete	cnt_C	payroll	Dec 3, 2003 5:59:14 PM	Payroll data
FCB_image	Incomplete	cnt_B	website	Dec 3, 2003 5:53:26 PM	Web site data
FCA_image	Complete	cnt_A	personnel	Dec 3, 2003 5:48:46 PM	Personnel data

## Islun

Lists the logical unit numbers (LUNs) that are accessible from SAN File System.



## Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

-fmt

Specifies the format of the output. You can specify one of the following values:

**default**

Specify this value to display output in a tabular format using spaces as the delimiter between the columns. This is the default value. For example:

OS Device Path	LUN ID	Engine WWN	Vendor	Product
/dev/sda	2	12:34:56:78:91:23:45:67	IBM	2105F20
Size (MB)	Volume	State		
47000	Vol1	Assigned		

**delim** *character*

Specify this value to display output in a tabular format using the specified character to separate the columns. If you use a shell metacharacter (for example, \* or \t) as the delimiting character, enclose the character in single quotation marks (') or double quotation marks ("). A blank space is not a valid character. For example:

```
/dev/sda,2,12:34:56:78:91:23:45:67,IBM,2105F20,47000,Vol1,Assigned
```

xml

Specify this value to display output using XML format, for example:

```
<RETURNVALUE>
<INSTANCE CLASSNAME="STC_AvailableLUNs">
<PROPERTY NAME="LUNID" TYPE="uint64"><VALUE TYPE="uint64">2</VALUE>
</PROPERTY>
<PROPERTY NAME="NodeWWN" TYPE="string"><VALUE TYPE="string">
12&58;34&58;56&58;78&58;91&58;23&58;45&58;67</VALUE></PROPERTY>
<PROPERTY NAME="Vendor" TYPE="string"><VALUE TYPE="string">IBM</VALUE>
</PROPERTY>
<PROPERTY NAME="Product" TYPE="string"><VALUE TYPE="string">2105F20
</VALUE></PROPERTY>
```

```

<PROPERTY NAME="Size" TYPE="uint64"><VALUE TYPE="uint64">47000</VALUE>
</PROPERTY>
<PROPERTY NAME="State" TYPE="uint64"><VALUE TYPE="uint64">2</VALUE>
</PROPERTY>
<PROPERTY NAME="VolumeName" TYPE="string"><VALUE TYPE="string"></VALUE>
</PROPERTY>
<PROPERTY NAME="DeviceID" TYPE="string"><VALUE TYPE="string">
&47;dev&47;sda</VALUE></PROPERTY>
</INSTANCE>
</IRETURNVALUE>

```

**-p** Specifies whether to display one page of text at a time or all text at once.

**off** Displays all text at one time. This is the default value when the **sfscli** command is run in single-shot mode.

**on** Displays one page of text at time. Pressing any key displays the next page. This is the default value when the **sfscli** command is run in interactive mode.

**-hdr**

Specifies whether to display the table header.

**on** Displays the table header. This is the default value.

**off** Does not display the table header.

**-r** *number*

Specifies the number of rows per page to display when the **-p** parameter is on. The default is 24 rows. You can specify a value from 1 to 100.

**-v** Specifies whether to enable verbose mode.

**off** Disables verbose mode. This is the default value.

**on** Enables verbose mode.

**-l** Displays the LUN IDs and other information about the LUNs.

**-s** Displays the LUNs by their LUN IDs.

**-client** *client\_name*

Displays information for only those data LUs that are visible to the specified clients as specified by SAN zoning. Any client that is zoned similarly has access to the same LUs.

This parameter reports the state of the LUN as assigned, if there is a SAN File System disk label on the LUN. It also reports the volume name to which this LUN is assigned. However if there is a valid SAN File System disk label on the LUN but this label does not belong to the current cluster, a question mark (?) is returned for the volume name. This can happen either because the LUN belongs to another SAN File System cluster, or because the disk label corresponds to a previous installation that did not remove the label. In either case, if a volume needs to be created on the LUN, you can specify the **-f** parameter to force the volume creation.

**-vendor** *vendor\_name*

Displays information for only those LUs that match the specified vendor name.

**-product** *product\_name*

Displays information for only those LUs that match the specified product name.

**-vol** *volume\_name*

Displays information for only those LUs that match the specified SAN File System volume name.

### **-state**

Displays information for only those LUs that match the specified state. Possible states are:

#### **assigned**

Displays only those LUs that are assigned to a storage pool and usable.

#### **available**

Displays only those LUs that are visible to SAN File System and usable but are not assigned to a storage pool.

**error** Displays only those LUs for which an error occurred while determining their availability.

#### **unknown**

Displays only those LUs whose availability could not be determined because the metadata server is not running.

#### **unusable**

Displays only those LUs that are unsuitable to be a volume. An LU is considered unsuitable if it is read-only, if its availability from all metadata servers in the cluster is inconsistent, or if the LU does not have an associated raw device.

### *lun\_id*

Specifies one or more identifiers of LUNs for which to display information.

- Specifies that you want to read one or more LUNs from stdin (for example,
  - << /work/device\_paths\_list.txt).

### **Prerequisites**

The cluster must not be in an offline state when you run this command.

### **Description**

This command operates only against the local engine.

By default, this command lists the metadata LUNs.

If you do not specify any filtering parameters, this command lists all the LUNs that are accessible to SAN File System, including LUNs that have not been added to SAN File System, SAN File System volumes, and unusable LUNs.

This command displays the following default information for each LUN:

- Logical unit number (LUN) identifier.
- Vendor.
- Product.
- Size of the device, in MB.
- Volume name, if the LUN is assigned in SAN File System.
- State of the LU.

### **Example**

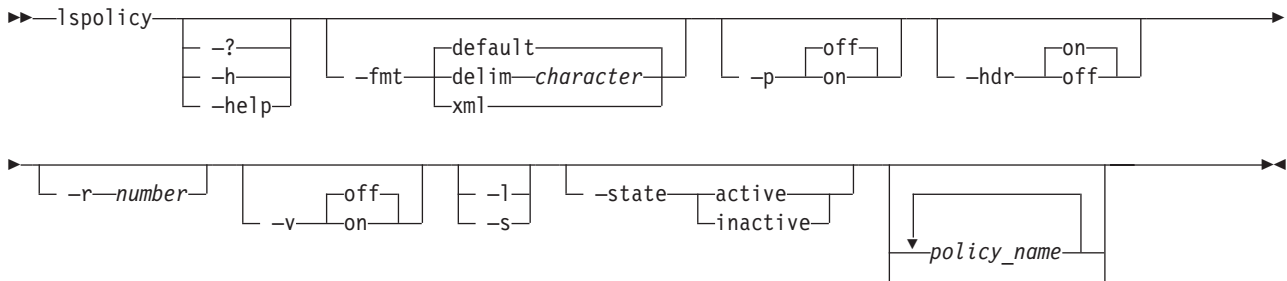
**Display LUNs** The following example lists all the attributes of all the available LUNs:

```
sfscli> ls1un -client client03
Lun ID Vendor Product Size (MB) Volume State
=====
VPD83Type1=IBM 2105 2CF24903 IBM 2105800 1907 - Available
```

VPD83Type1=IBM	2105	2CE24903 IBM	2105800	1907 -	Available
VPD83Type1=IBM	2105	2CC24903 IBM	2105800	1907 -	Available
VPD83Type1=IBM	2105	2CD24903 IBM	2105800	1907 -	Available
VPD83Type1=IBM	2105	2CB24903 IBM	2105800	1907 -	Available
VPD83Type1=IBM	2105	2CA24903 IBM	2105800	1907 -	Available
VPD83Type1=IBM	2105	2C924903 IBM	2105800	1907 -	Available
VPD83Type1=IBM	2105	2BD24903 IBM	2105800	4768 MASTER	Assigned
VPD83Type1=IBM	2105	3B624903 IBM	2105800	4768 -	Available
VPD83Type1=IBM	2105	3B524903 IBM	2105800	4768 -	Available
VPD83Type1=IBM	2105	2D324903 IBM	2105800	1907 -	Available

## lspolicy

Lists the active and inactive policies.



### Parameters

#### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### -fmt

Specifies the format of the output. You can specify one of the following values:

##### default

Specifies to display output in a tabular format using spaces as the delimiter between the columns. This is the default value. For example:

Name	State	Last Active	Modified
policy1	Active	Jul 19, 2003 14:00:34 PM	Jul 19, 2003 13:59:34 PM

Description  
=====

Active policy

##### delim *character*

Specifies to display output in a tabular format using the specified character to separate the columns. If you use a shell metacharacter (for example, \* or \t) as the delimiting character, enclose the character in single quotation marks (') or double quotation marks ("). A blank space is not a valid character. For example:

policy1,Active,Jul 19, 2003 14:00:34 PM,Jul 19, 2003 13:59:34 PM,Active policy

##### xml

Specifies to display output using XML format, for example:

```
<IRETURNVALUE>
<INSTANCE CLASSNAME="STC_PolicySet">
<PROPERTY NAME="Name" TYPE="string"><VALUE TYPE="string">policy1</VALUE>
</PROPERTY>
<PROPERTY NAME="State" TYPE="uint16"><VALUE TYPE="uint16">1</VALUE>
```

```

</PROPERTY>
<PROPERTY NAME="Description" TYPE="string"><VALUE TYPE="string">
Active policy</VALUE></PROPERTY>
<PROPERTY NAME="LastModificationDate" TYPE="datetime"><VALUE
TYPE="datetime">20030719135934-000000-420</VALUE></PROPERTY>
<PROPERTY NAME="LastActiveDate" TYPE="datetime"><VALUE TYPE="datetime">
20030719140034-000000-420</VALUE></PROPERTY>
</INSTANCE>
</IRETURNVALUE>

```

**-p** Specifies whether to display one page of text at a time or all text at once.

**off** Displays all text at one time. This is the default value when the **sfscli** command is run in single-shot mode.

**on** Displays one page of text at time. Pressing any key displays the next page. This is the default value when the **sfscli** command is run in interactive mode.

**-hdr**

Specifies whether to display the table header.

**on** Displays the table header. This is the default value.

**off** Does not display the table header.

**-r** *number*

Specifies the number of rows per page to display when the **-p** parameter is on. The default is 24 rows. You can specify a value from 1 to 100.

**-v** Specifies whether to enable verbose mode.

**off** Disables verbose mode. This is the default value.

**on** Enables verbose mode.

**-l** Displays the default information about each policy.

**-s** Displays only the name of each policy.

**-state** **active** | **inactive**

Specifies the state of the policies to display. You can specify one of the following values:

**active** Displays only the active policy.

**inactive**

Displays only the inactive policies.

*policy\_name*

Specifies the names of one or more policies to list.

**-** Specifies that you want to read the names of one or more policies to list from stdin (for example, **- << /work/policies\_list.txt**).

## Prerequisites

You must be logged in to the engine hosting the master metadata server to run this command.

## Description

When you specify multiple parameters in this command, these behaviors apply:

- Flags work as AND coordinators.
- Objects work as OR coordinators.

- When flags and objects are combined, the they are listed in this manner:  
WHERE flag\_1 AND flag\_2 AND ... AND flag\_n AND (object\_1 OR object\_2 OR ... OR object\_m).
- When flags or objects are not required and not specified, this command searches all possible values unless a default overrides this rule.

For example, if you specify `-state` and two policy names, the search would be performed in the following manner: WHERE state AND (policy\_name\_1 OR policy\_name\_2).

If you do not specify a listing parameter or if you specify the `-l` listing parameter, this command displays the following information for each policy:

- Name of the policy.
- State of the policy.
- Timestamp when the policy was last active, if applicable, based on coordinated universal time (UTC).
- Timestamp when the policy was last modified, based on coordinated universal time (UTC).
- Policy description.

### Example

**Display policies** The following example displays all the available information about all the policies:

```
sfsccli> lspolicy -l
```

Name	State	Last Active	Modified
policy1	Active	Jul 19, 2003 14:00:34 PM	Jul 19, 2003 13:59:34 PM
testpolicy	Inactive	Mar 18, 2003 10:30:17 PM	Mar 18, 2003 10:06:25 PM
DEFAULT	Inactive	Jan 11, 2003 16:36:24 PM	Jan 11, 2003 15:36:14 PM

```

Description
=====
Active policy
Test policy
Default Policy

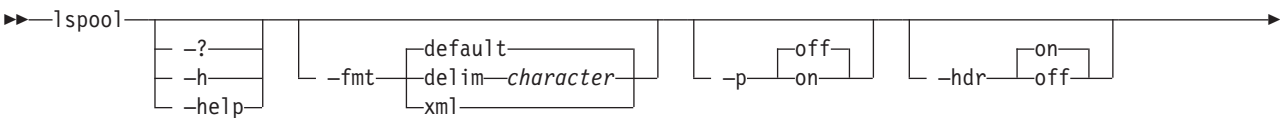
```

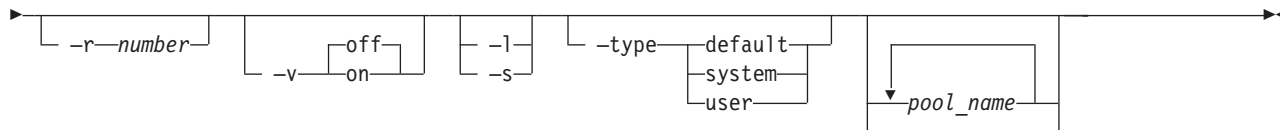
**Display the rules of the active policy** The following example displays the rules of the active policy:

```
$sfsccli lspolicy -s -state active -hdr off | sfsccli catpolicy -
VERSION 1
rule 'stgRule1' set stgpool 'pool1' for fileset ('cnt_A')
rule 'stgRule2' set stgpool 'pool2' where NAME like '%.doc'
rule 'stgRule3' set stgpool 'pool3' where DAYOFWEEK(CREATION_DATE) == 1
rule 'stgRule4' set stgpool 'pool4' where USER_ID <= 100
```

## lspool

Displays a list of the existing storage pools and their attributes.





## Parameters

### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### -fmt

Specifies the format of the output. You can specify one of the following values:

#### default

Specify this value to display output in a tabular format using spaces as the delimiter between the columns. This is the default value. For example:

Name	Type	Size (MB)	Used (MB)	Used (%)	Threshold (%)	Volumes
Pool1	User	1400	575	41	90	2

#### delim *character*

Specify this value to display output in a tabular format using the specified character to separate the columns. If you use a shell metacharacter (for example, \* or \t) as the delimiting character, enclose the character in single quotation mark (') or double quotation mark ("). A blank space is not a valid character. For example:

```
Pool1,User,
```

#### xml

Specify this value to display output using XML format, for example:

```
<IRETURNVALUE>
<INSTANCE CLASSNAME="STC_StoragePool">
<PROPERTY NAME="Name" TYPE="string"><VALUE TYPE="string">Pool1</VALUE>
</PROPERTY>
<PROPERTY NAME="PoolType" TYPE="uint32"><VALUE TYPE="uint32">0</VALUE>
</PROPERTY>
<PROPERTY NAME="AlertPercentage" TYPE="uint16"><VALUE TYPE="uint16">90
</VALUE></PROPERTY>
<PROPERTY NAME="Size" TYPE="uint64"><VALUE TYPE="uint64">1400</VALUE>
</PROPERTY>
<PROPERTY NAME="SizeAllocated" TYPE="uint64"><VALUE TYPE="uint64">575
</VALUE></PROPERTY>
<PROPERTY NAME="SizeAllocatedPercentage" TYPE="uint16"><VALUE
TYPE="uint16">41</VALUE></PROPERTY>
<PROPERTY NAME="NumberOfVolumes" TYPE="uint32"><VALUE TYPE="uint32">2
</VALUE></PROPERTY>
</INSTANCE>
</IRETURNVALUE>
```

### -p

- off** Displays all text at one time. This is the default value when the **sfscli** command is run in single-shot mode.
- on** Displays one page of text at time. Pressing any key displays the next page. This is the default value when the **sfscli** command is run in interactive mode.



- hdr**  
Specifies whether to display the table header.
- on** Displays the table header. This is the default value.
  - off** Does not display the table header.
- r *number***  
Specifies the number of rows per page to display when the **-p** parameter is on. The default is 24 rows. You can specify a value from 1 to 100.
- v** Specifies whether to enable verbose mode.
- off** Disables verbose mode. This is the default value.
  - on** Enables verbose mode.
- l** Displays additional information about the storage pools.
- s** Displays only the names of the storage pools. This is the default.
- type**  
Displays information for only those storage pools that match the specified type. If not specified, all storage pools are displayed. Possible types are:
- default** Displays information for the default storage pool.
  - system** Displays information for the system storage pool.
  - user** Displays information for all user storage pools.
- pool\_name***  
Specifies the names of one or more storage pools to display.
- Specifies that you want to read the names of one or more storage pools to display from stdin (for example, - << /work/stgpools\_list.txt).

### Prerequisites

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

### Description

If you do not specify a listing parameter, this command displays the following default information for each storage pool:

- Name of the storage pool.
- Type of storage pool.
- Size of the storage pool, in MB.
- Amount of space used, in MB.
- Percent of space used.
- Alert threshold.
- Number of volumes in the storage pool.

If you specify the **-l** listing parameter, this command displays the following information in addition to the default information for each storage pool:

- Partition size, in MB (16, 64, or 256).
- Allocation size, in KB (auto, 4 or 128).
- Pool description.

## Example

**Display a long listing** The following example lists all the attributes of all the storage pools:

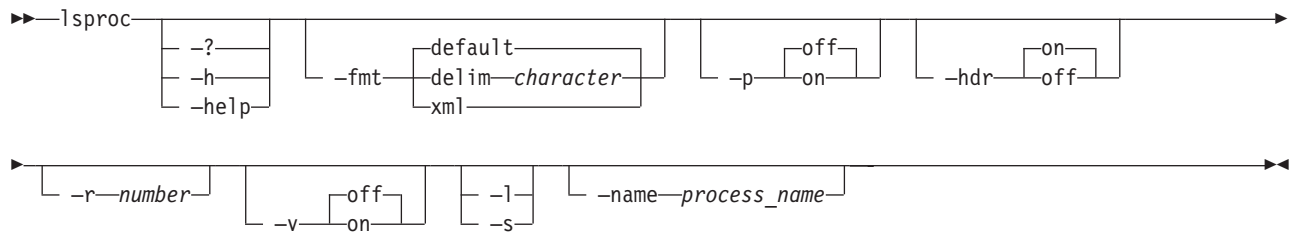
```
sfscli> lspool -l
```

Name	Type	Size (MB)	Used (MB)	Used (%)	Threshold (%)	Volumes
Pool1	User	1400	575	41	90	2
Pool2	User	2000	1000	50	90	3
DEFAULT	Default	10000	2500	25	80	10
SYSTEM	System	10000	7500	75	80	10

Partition Size (MB)	Allocation Size (KB)	Description
16	2	Accounting
16	3	Engineering
64	128	DEFAULT pool
64	128	SYSTEM pool

## lsproc

Displays a list of long-running processes that are not yet complete and their attributes.



### Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

**-fmt**

Specifies the format of the output. You can specify one of the following values:

**default**

Specifies to display output in a tabular format using spaces as the delimiter between the columns. This is the default value. For example:

```
ID Name Started
=====
5 CHECKMETADATA Feb 11, 2003 9:29:56:13 PM
```

**delim character**

Specifies to display output in a tabular format using the specified character to separate the columns. If you use a shell metacharacter (for example, \* or \t) as the delimiting character, enclose the character in single quotation mark (') or double quotation mark ("). A blank space is not a valid character. For example:

```
5,CHECKMETADATA, Feb 11, 2003 9:29:56:13 PM
```

**xml**

Specifies to display output using XML format, for example:

```

<IRETURNVALUE>
<INSTANCE CLASSNAME="STC_AdminProcesses">
<PROPERTY NAME="Id" TYPE="uint64"><VALUE TYPE="uint64">5</VALUE>
</PROPERTY>
<PROPERTY NAME="Command" TYPE="string"><VALUE TYPE="uint32">STOPSERVER
</VALUE></PROPERTY>
<PROPERTY NAME="InstallDate" TYPE="datetime"><VALUE TYPE="datetime">
20030211092956.000000-420</VALUE></PROPERTY>
</INSTANCE>
</IRETURNVALUE>

```

**-p** Specifies whether to display one page of text at a time or all text at once.

**off** Displays all text at one time. This is the default value when the **sfscli** command is run in single-shot mode.

**on** Displays one page of text at time. Pressing any key displays the next page. This is the default value when the **sfscli** command is run in interactive mode.

**-hdr**

Specifies whether to display the table header.

**on** Displays the table header. This is the default value.

**off** Does not display the table header.

**-r** *number*

Specifies the number of rows per page to display when the **-p** parameter is on. The default is 24 rows. You can specify a value from 1 to 100.

**-v** Specifies whether to enable verbose mode.

**off** Disables verbose mode. This is the default value.

**on** Enables verbose mode.

**-l** Displays additional information about the processes.

**-s** Displays only the identifier of each process.

**-name** *process\_name*

Specifies the name of a process to list.

## Prerequisites

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

If you do not specify a listing parameter or if you specify the **-l** listing parameter, this command displays the following information for each process:

- Identifier of the process.
- Name of the process.
- Time that the process was started.

## Example

**Lists long-running processes** The following example lists the long-running processes currently in progress:

```
sfscli> lsproc
```

```

ID Name Started
=====

```

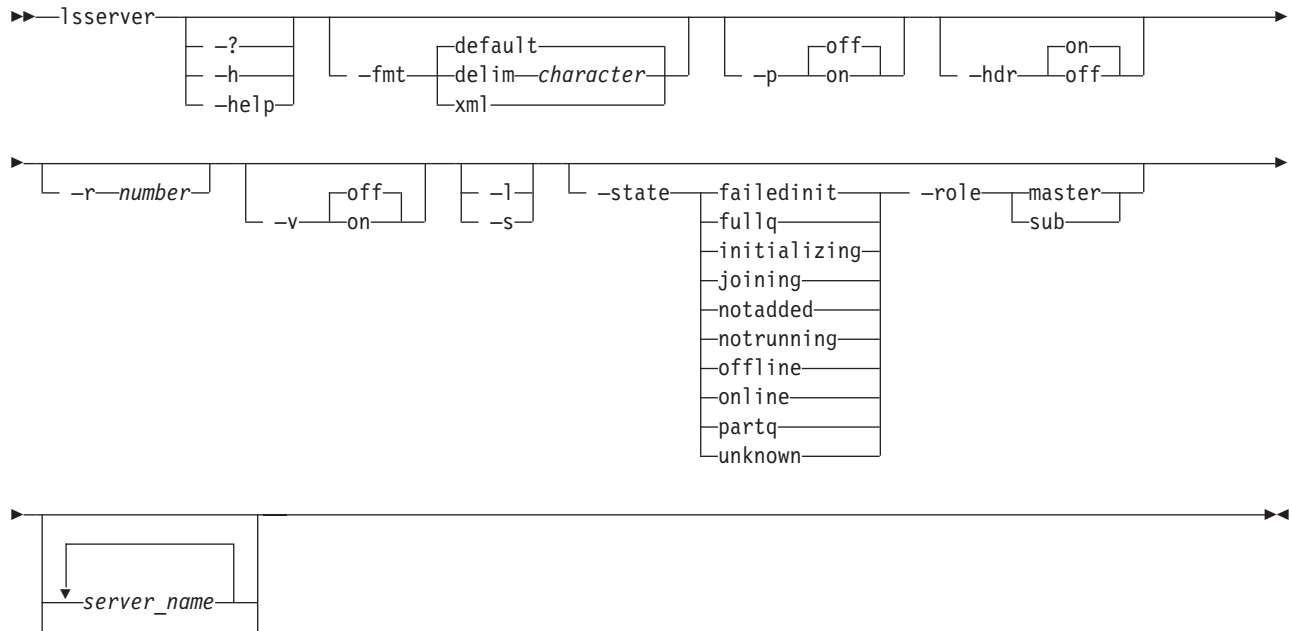
```

5 CHECKMETADATA Feb 11, 2003 9:29:56:13 PM
7 STOPCLUSTER Feb 10, 2003 9:30:20:24 PM
9 STARTSERVER Feb 10, 2003 9:12:22:24 PM

```

## Isserver

Displays a list of all metadata servers in the cluster and their attributes if issued from the master metadata server, or displays attributes about the local metadata server if issued from a subordinate metadata server.



### Parameters

#### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### -fmt

Specifies the format of the output. You can specify one of the following values:

##### default

Specify this value to display output in a tabular format using spaces as the delimiter between the columns. This is the default value. For example:

Name	State	Relationship	Filesets	Last Boot
ST1	ONLINE	Subordinate	3	Dec. 31, 2002 5:00:34 PM

##### delim character

Specify this value to display output in a tabular format using the specified character to separate the columns. If you use a shell metacharacter (for example, \* or \t) as the delimiting character, enclose the character in single quotation mark (') or double quotation mark ("). A blank space is not a valid character. For example:

```
ST1;ONLINE;Subordinate;3;Dec. 31, 2002 5:00:34 PM
```

##### xml

Specify this value to display output using XML format, for example:

```

<IRETURNVALUE>
<INSTANCE CLASSNAME="STC_TankService">
<PROPERTY NAME="CurrentState" TYPE="uint32"><VALUE TYPE="uint32">1
</VALUE></PROPERTY>
<PROPERTY NAME="LastBootUpTime" TYPE="datetime"><VALUE TYPE="datetime">
20031231050034.000000-420</VALUE></PROPERTY>
<PROPERTY NAME="SystemName" TYPE="string"><VALUE TYPE="string">
DEFAULT_POOL</VALUE>ST1</PROPERTY>
<PROPERTY NAME="IsMaster" TYPE="boolean"><VALUE TYPE="boolean">>false
</VALUE></PROPERTY>
<PROPERTY NAME="NumberOfFilesets" TYPE="uint32"><VALUE TYPE="uint32">3
</VALUE></PROPERTY>
</INSTANCE>
</IRETURNVALUE>

```

**-p** Specifies whether to display one page of text at a time or all text at once.

**off** Displays all text at one time. This is the default value when the **sfscli** command is run in single-shot mode.

**on** Displays one page of text at time. Pressing any key displays the next page. This is the default value when the **sfscli** command is run in interactive mode.

**-hdr**

Specifies whether to display the table header.

**on** Displays the table header. This is the default value.

**off** Does not display the table header.

**-r number**

Specifies the number of rows per page to display when the **-p** parameter is on. The default is 24 rows. You can specify a value from 1 to 100.

**-v** Specifies whether to enable verbose mode.

**off** Disables verbose mode. This is the default value.

**on** Enables verbose mode.

**-l** Displays additional information about the metadata servers.

**-s** Displays only the name of each metadata server. This is the default.

**-state**

Lists information for only those metadata servers that match the specified state:

**failedinit**

Displays information for metadata servers with the state Failed Initialization. A fatal error occurred during initialization and the metadata server is suspended. The metadata server remains suspended so you can correct the problem that occurred during initialization.

**fullq**

Displays information for metadata servers with the state Fully Quiescent. The metadata server suspends all current client sessions and prevents new client sessions from being started. All file-data and metadata activity is suspended. The metadata server flushes all dirty cached pages and commits them to disk. This state is used to produce self-consistent LUN-based backups from a known point-in-time.

**initializing**

Displays information for metadata servers with the state Initializing. The metadata server is running, but has not yet opened its communication ports.

**joining**

Displays information for metadata servers with the state Joining. The metadata server is joining the cluster.

**notadded**

Displays information for metadata servers with the state Not Added. The metadata server is not part of the cluster. Because the master metadata server does not know about metadata servers that are not part of the cluster, this state is only available by logging in to that metadata server.

**notrunning**

Displays information for metadata servers with the state NOT RUNNING. The metadata server is part of the cluster but is not running and cannot perform any functions.

**offline**

Displays information for metadata servers with the state OFFLINE. The metadata server ends all current client sessions and prevents new client sessions from being started. This state allows all metadata server I/O to continue. This state restricts the metadata server from client access.

**online** Displays information for metadata servers with the state ONLINE. The metadata server has fully initialized all of its subsystems, is a member of the cluster, and is serving client requests.

**partq** Displays information for metadata servers with the state PARTLY QUIESCENT. The metadata server prevents new client sessions from being started but allows file I/O to continue for existing clients. The metadata server suspends clients making file metadata updates (such as changing permissions or creating new files) and system metadata activity. The metadata server also flushes any dirty metadata pages in the client and metadata server caches. This state allows you to produce dirty LUN-based backups without requiring clients and applications to stop. You can back up applications using direct I/O (such as database servers) (no caching) in this state.

**unknown**

Displays information for metadata servers with the state UNKNOWN.

**-role master | sub**

Displays only information for those metadata servers that match the specified role:

**master**

Displays information for the master metadata server.

**sub**

Displays information for subordinate metadata servers.

***server\_name***

Specifies the name of one or more names of metadata servers to display. If not specified and if this command is issued from the master issued from the master metadata server, lists information about all metadata servers in the cluster.

- Specifies that you want to read the names of one or more of the metadata servers to display from stdin (for example, - << /work/server\_list.txt).

## Description

If you run this command from an engine hosting a subordinate metadata server, you can display information about only the local metadata server. If you run this command from the engine hosting the master metadata server, you can display information about any metadata server.

When you specify multiple parameters in this command, these behaviors apply:

- Flags work as AND coordinators.
- Objects work as OR coordinators.
- When flags and objects are combined, they are listed in this manner:  
`WHERE flag_1 AND flag_2 AND ... AND flag_n AND (object_1 OR object_2 OR ... OR object_m)`
- When flags or objects are not required and not specified, this command searches all possible values unless a default overrides this rule.

For example, if you specify `-state`, `-role`, and two metadata server names, the search would be performed in the following manner:

`WHERE state AND role AND (server_name_1 OR server_name_2)`

If you do not specify a listing parameter, this command displays the following default information for each metadata server:

- Name of the metadata server.
- State of the metadata server.
- Role of the metadata server (master or subordinate).
- Number of filesets that are hosted by the metadata server.
- Timestamp of the last boot.

If you specify the `-l` listing parameter, this command displays the following information in addition to the default information for each metadata server:

- Timestamp of the last status change.
- State to which the metadata server changed.
- Timestamp of the last target state change.
- Current time.
- Pending software version.

## Example

**Display online metadata servers** The following example lists all the attributes of all the metadata servers with the state ONLINE:

```
sfsccli> lsserver -state online
```

Name	State	Relationship	Filesets	Last Boot
ST1	ONLINE	Subordinate	3	Dec. 31, 2002 5:00:34 PM
ST2	ONLINE	Master	2	Dec. 31, 2002 5:00:24 PM

## lssnmpmgr

Displays a list of SNMP managers and their attributes.



### Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### Prerequisites

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

### Description

This command displays the following information for each SNMP manager:

- SNMP manager identifier.
- IP address.
- Port number.
- Version.
- Community.

### Example

**Lists the SNMP managers** The following example lists the SNMP managers:

```
sfsccli> lssnmpmgr
```

ID	IP	Port	Version	Community
1	192.168.0.1	8192	V1	Public
2	192.168.0.2	162	V2C	SNMPMgr2

## lstrapsetting

Displays a list of event types that currently generate an SNMP trap.



### Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.



Prerequisites

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

Description

This command displays the whether the following event types are enabled:

- Information
- Error
- Warning
- Severe

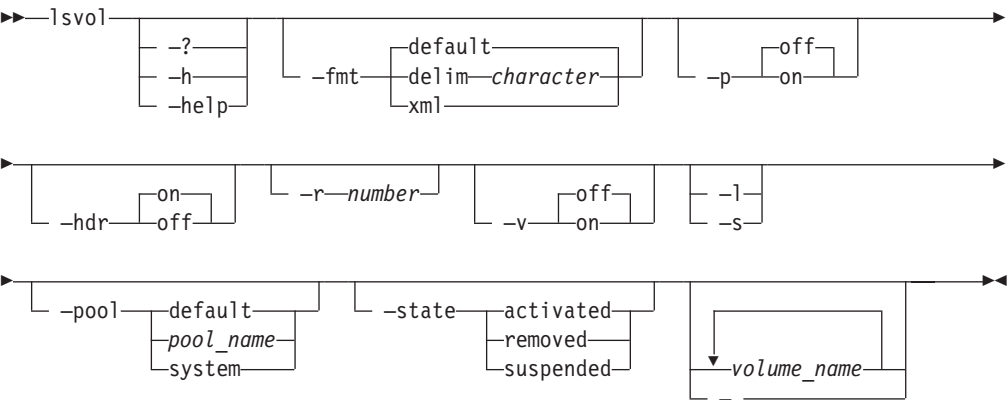
Example

**Displays which event types generate SNMP traps** The following example shows that SNMP traps are generated for severe and warning events:

```
sfscli> lstrapsetting
Information Disabled
Error Disabled
Warning Enabled
Severe Enabled
```

lsvol

Displays a list of available volumes.



Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

**-fmt**

Specifies the format of the output. You can specify one of the following values:

**default**

Specifies to display output in a tabular format using spaces as the delimiter between the columns. This is the default value. For example:

Name	State	Pool	Size (MB)	Used (MB)	Used (%)
Vol1	Activated	P_Accting	700	350	50

**delim** *character*

Specifies to display output in a tabular format using the specified character to separate the columns. If you use a shell metacharacter (for example, \* or \t) as the delimiting character, enclose the character in single quotation marks (') or double quotation marks ("). A blank space is not a valid character. For example:

Vol1,Activated,P\_Accting,700,350,50

**xml** Specifies to display output using XML format, for example:

```
<IRETURNVALUE>
<INSTANCE CLASSNAME="STC_Volume">
<PROPERTY NAME="StoragePoolName" TYPE="string"><VALUE TYPE="string">
P_Accting</VALUE></PROPERTY>
<PROPERTY NAME="Name" TYPE="string"><VALUE TYPE="string">Vol1</VALUE>
</PROPERTY>
<PROPERTY NAME="State" TYPE="uint32"><VALUE TYPE="uint32">0</VALUE>
</PROPERTY>
<PROPERTY NAME="Size" TYPE="uint64"><VALUE TYPE="uint64">700</VALUE>
</PROPERTY>
<PROPERTY NAME="SizeAllocated" TYPE="uint64"><VALUE TYPE="uint64">350
</VALUE></PROPERTY>
<PROPERTY NAME="SizeAllocatedPercentage" TYPE="uint16"><VALUE
TYPE="uint16">50</VALUE></PROPERTY>
</INSTANCE>
</IRETURNVALUE>
```

**-p** Specifies whether to display one page of text at a time or all text at once.

**off** Displays all text at one time. This is the default value when the sfscli command is run in single-shot mode.

**on** Displays one page of text at time. Pressing any key displays the next page. This is the default value when the sfscli command is run in interactive mode.

**-hdr**

Specifies whether to display the table header.

**on** Displays the table header. This is the default value.

**off** Does not display the table header.

**-r** *number*

Specifies the number of rows per page to display when the **-p** parameter is on. The default is 24 rows. You can specify a value from 1 to 100.

**-v** Specifies whether to enable verbose mode.

**off** Disables verbose mode. This is the default value.

**on** Enables verbose mode.

**-l** Displays additional information about the volumes.

**-s** Displays only the name of the volumes.

**-pool** *pool\_name*

Specifies the name of the storage pool to display. You can specify one of the following values:

- **default** (for the default storage pool)
- User-defined storage pool name
- **system** (for the system storage pool)

**-state**

Displays only information for those storage pools that match the specified type. Possible types are:

**activated**

Displays information for all volumes with a status of Activated.

**removed**

Displays information for all volumes with a status of Being Removed.

**suspended**

Displays information for all volumes with a status of Suspended.

*volume\_name*

Specifies the names of one or more volumes to list. If unspecified, displays all volume names.

- Specifies that you want to read the names of one or more volumes to list from stdin (for example, - << /work/vol\_list.txt).

**Prerequisites**

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

**Description**

When you specify multiple parameters in this command, these behaviors apply:

- Flags work as AND coordinators.
- Objects work as OR coordinators.
- When flags and objects are combined, they are listed in this manner:  
WHERE *flag\_1* AND *flag\_2* AND ... AND *flag\_n* AND (*object\_1* OR *object\_2* OR ... OR *object\_m*).
- When flags or objects are not required and not specified, this command searches all possible values unless a default overrides this rule.

For example, if you specify -pool, -state, and two volume names, the search would be performed in the following manner: WHERE *pool* AND *state* AND (*volume\_name\_1* OR *volume\_name\_2*).

If you do not specify a listing parameter, this command displays the following default information for each volume:

- The name of the volume.
- The state of the volume.
- The size of the volume, in MB.

**Note:** When the metadata server creates a volume, it uses some space for its internal bookkeeping (including the disk label). This command reports the size of the volume as the amount of space that can be used to store data, which does not include the space used for bookkeeping.

- The amount of space being used, in MB.
- The percent of space being used.

If you specify the -l listing parameter, this command displays the following information in addition to the default information for each volume:

- The operating-system device path.
- The SAN File System volume ID.

- The volume description.

### Example

**Display a list of volumes** The following example lists all the attributes of all the volumes:

```
sfscli> lsvol -l
```

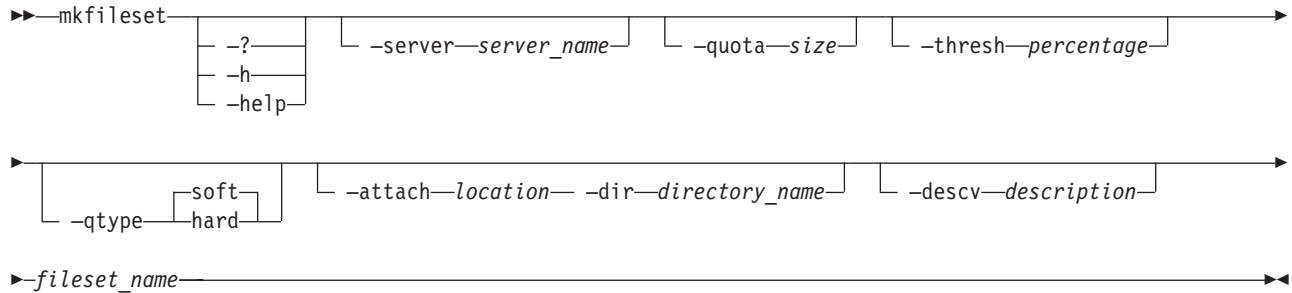
Name	State	Pool	Size (MB)	Used (MB)	Used (%)
Vol1	Activated	P_Accting	700	350	50
Vol2	Being Removed	P_Accting	700	175	25

Device Path	Vol ID	Description
/dev/hrvpatha	912478	Department 1
/dev/hrvpathb	807890	Department 2

## mkfileset

Creates a new fileset.



### Parameters

#### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### -server *server\_name*

Specifies the metadata server to host the fileset. This name can be up to 32 characters in length.

- If you do not specify a metadata server, the fileset is assigned to a metadata server automatically.
- If the specified metadata server goes down, the fileset is reassigned to another metadata server automatically.

This metadata server can be in any state as long as it is part of the cluster.

#### -quota *size*

Specifies the maximum size limit, in megabytes, for the specified fileset that, when exceeded, causes the metadata server to generate an alert. You can specify a number from 0 to 1 073 741 824 MB (1 PB). The default size is 0.

If not specified or set to 0, there is no quota size limit for this fileset, and alerts are not sent.

#### -thresh *percentage*

Specifies the maximum percentage (alert threshold) of the specified quota size

for this fileset that, when exceeded, causes the metadata server to generate an alert. You can specify a value between 0 and 100. The default alert percentage is 80.

If the **-thresh** parameter is set to 0 or if the quota size is not specified, no alerts are generated.

**-qtype hard | soft**

Specifies the quota type for the fileset. You can specify one of the following values:

**hard** A hard quota produces a log message and potential alert when the quota is met, and denies requests for more space.

**soft** A soft quota produces a log message and potential alert when the quota size is exceeded, but grants requests for more space. This is the default type.

**-attach *location***

Specifies the existing directory path (relative to the root of the global namespace and without the / prefix) at which to attach the fileset.

The root of the global namespace must be included in the directory path. For example, to attach the fileset to the /homes location, specify **-attach sanfs/homes**. To attach the fileset directly to the root of the global namespace, use **-attach sanfs**.

**Requirements:** This directory must already exist. Use only forward slashes (/) in the directory path for delimiters.

**Tip:** If the **-attach** parameter is not specified, the fileset cannot be attached to the global namespace.

**-dir *directory\_name***

Specifies the directory name for the root of the fileset directory tree. This directory can be up to 256 characters in length.

The attach point (*location/directory\_name*) appears as a drive and directory on Windows® clients or as a file system on UNIX®-based clients.

**Tip:** Map the name of the directory to its associated fileset by using the same name as the fileset plus an additional suffix to distinguish it as the directory and not the fileset. For example, if you want to attach the fileset named work to the root of the global namespace, specify **sanfs** for the *location* and specify **work\_ap** for the *directory\_name*. On a Windows client, the work\_ap directory appears under the sanfs drive.

**Requirements:** The directory name must not already exist. The directory must not contain backslash (\) or forward slash (/) characters as delimiters.

**-desc *description***

Specifies a description for the fileset. The description must be enclosed in matching single (') or double (") quotation marks if it contains any blank characters. This description can be up to 256 characters in length.

The default is an empty string.

***fileset\_name***

Specifies the name to assign to the new fileset. This name can be up to 256 characters in length.

## Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

You must attach the fileset to the global namespace using the **-attach** and **-dir** parameters in this command or the **attachfileset** command before clients can access the fileset.

Newly created filesets have permissions set to the following values:

- 555 permissions when viewed from UNIX-based clients.
- Read control (to Everyone) and Write discretionary access control list (DACL)/Owner (for administrator or root clients) from Windows-based clients.

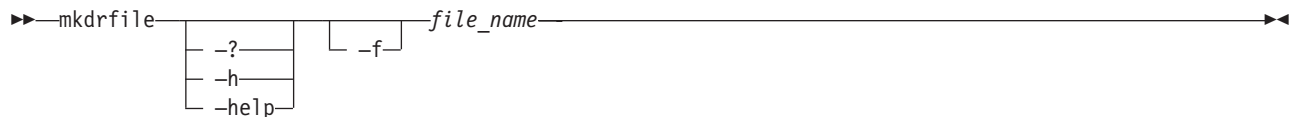
## Example

**Create filesets** The following example creates a fileset named `cnt_A`, assigns the fileset to the ST1 metadata server, sets the quota to 1 000 MB, specifies to send alerts when the quota reaches 70%, and attaches the fileset to the `sanfs/homes` directory.

```
sfscli> mkfileset -server ST1 -quota 1000 -thresh 70 -qtype soft
-attach sanfs/homes -dir cntA_ap cnt_A
Fileset "cnt_A" created.
```

## mkdrfile

Creates a new system-metadata disaster-recovery dump file.



## Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

**-f** Overwrites the system-metadata disaster-recovery dump file if a file with the specified name already exists.

*file\_name*

Specifies the name of the new system-metadata disaster-recovery dump file. This name can be up to 250 characters in length. The created file can be found in the `/usr/tank/server/DR` directory.

## Prerequisites

You must have Backup, Operator, or Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

This command is used in conjunction with the **bulddrscript** command to create a set of recovery scripts that are used to recreate the system metadata in the event of a disaster. The **bulddrscript** command is run against the system-metadata disaster-recovery file created by the **mkdrfile** command.

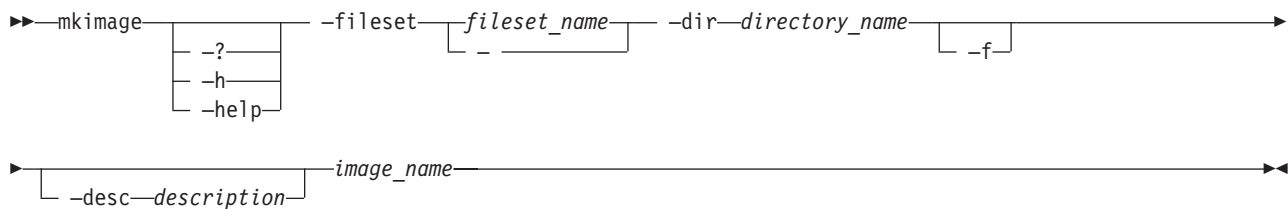
## Example

**Create a dump file** The following example creates a disaster-recovery dump file named “dr1”:

```
sfsccli> mkdrfile dr1
Disaster recovery file "dr1" was created successfully.
```

## mkimage

Creates a near-instantaneous FlashCopy image of the file layout and contents of the specified fileset and stores it in that fileset.



## Parameters

### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### -fileset fileset\_name | -

Specifies the name of the fileset for which to create a FlashCopy image. Specifying a - indicates that you want to read the name of the fileset from stdin (for example, - << /work/cnt\_list.txt).

### -dir directory\_name

Specifies the file-system directory under which you want to access the FlashCopy image. This directory can be up to 256 characters in length and must not contain backslash (\) or forward slash (/) characters as delimiters.

The directory name is created under the .flashcopy directory in the fileset. The full path of any file in the FlashCopy image is under the directory *attach\_point/.flashcopy/directory*, where *attach\_point* is the location to which the fileset attaches in the global namespace (for example, *sanfs/cnt\_A/.flashcopy/fc1dir*).

**Note:** The .flashcopy directory is a hidden directory. Windows-based clients must use special operating-system commands to view this directory.

**-f** Forces the metadata server to delete the oldest FlashCopy image if the fileset

already contains the maximum number of FlashCopy images. The oldest image is deleted even if the image is currently in use.

**Attention:** Any open files associated with the oldest image will be lost.

**-desc** *description*

Specifies a description for the FlashCopy image. The description must be enclosed in matching single (') or double (") quotation marks if it contains any blank characters. This description can be up to 256 characters in length.

The default is an empty string.

*image\_name*

Specifies the name of the FlashCopy image to create. This name must be unique within the fileset and can be up to 256 characters in length.

**Note:** This name must not contain colons (:).

### Prerequisites

You must have Backup, Operator, or Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

### Description

This command does not create FlashCopy images of nested filesets. You must create a FlashCopy image for each fileset in the hierarchy.

After you create a FlashCopy image, you cannot modify the name, description, or directory.

Until the **mkimage** command completes, you can view files in the fileset but not modify them.

You cannot use the **mkimage** command while the **reverttoimage** command is in progress on the same fileset.

A fileset is limited to 32 FlashCopy images at one time.

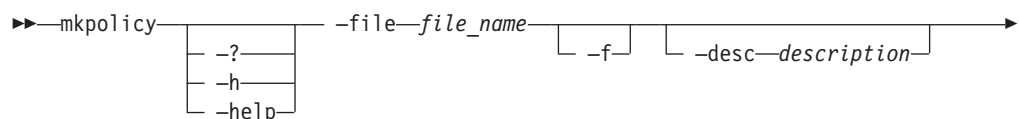
### Example

**Create a FlashCopy image** The following example creates a FlashCopy image for the fileset cnt\_A:

```
sfscli> mkimage -fileset cnt_A -dir website cntA_image
Image "cntA_image" successfully created.
```

## mkpolicy

Creates a policy.





► *policy\_name* ◀

## Parameters

### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### -file *file\_name*

Specifies the fully qualified path and name of the file that contains the set of rules to be included in the new policy. The maximum size of this file is 32 KB.

Use the following format to add rules to this file:

VERSION 1

```
rule 'stgRule1' set stgpool 'pool1' for fileset ('cnt_A')
rule 'stgRule2' set stgpool 'pool2' where NAME like '%.doc'
rule 'stgRule3' set stgpool 'pool3' where DAYOFWEEK(CREATION_DATE) == 1
rule 'stgRule4' set stgpool 'pool4' where USER_ID <= 100
```

### -f Forces the metadata server to overwrite an existing policy with the same name as the specified policy.

### -desc *description*

Specifies a description for the policy. The description must be enclosed in matching single (') or double (") quotation marks if it contains any blank characters. This description can be up to 256 characters in length.

The default is an empty string.

### *policy\_name*

Specifies the names of the policy to create. This name can be up to 256 characters in length.

- Specifies that you want to read the names of the policy to create from stdin (for example, - << /work/policies\_list.txt).

## Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

Each policy can be up to 32 KB.

To update the policy rules or attributes through the administrative command-line interface, you must create a new policy.

To edit the rules for an existing policy to use as input to create a new policy, send the output of the **catpolicy** command for the existing policy to a file and then edit that file. Use the edited file as the input file to the **mkpolicy** command to create the new policy.

See Appendix D, “File placement policy syntax,” on page 323 for a description of the syntax conventions for file-placement rules.

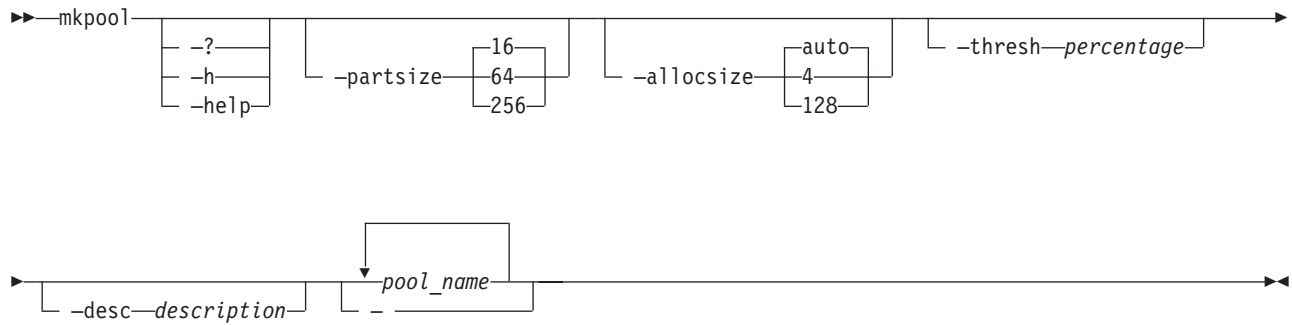
## Example

**Create a policy** The following example creates a policy (**test\_policy**) using the rules in the `/tmp/my_rules.txt` file:

```
sfscli> mkpolicy -file /tmp/my_rules.txt -desc "Test Policy" test_policy
Policy test_policy created.
```

## mkpool

Defines one or more new user storage pools.



### Parameters

#### `-? | -h | -help`

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### `-partsize size`

Specifies the partition size, in megabytes, to use when a fileset allocates space. The partition size can be set to 16, 64, or 256 MB. The default size is 16 MB.

**Restriction:** You *cannot* change the partition size after it is set.

For normal use, set the partition size to 16MB. To improve performance, especially for handling extremely large files, you can specify a larger partition size. Be aware that when you set a partition size higher than 16 MB, the pools take up more storage.

#### `-allocsize size`

Specifies the allocation size, in kilobytes, by which a file that resides on a storage pool is to be extended. The allocation size can be set to auto, 4, or 128. If you specify auto, the system sets the size automatically. The default value is auto.

**Restriction:** You cannot change the allocation size after it is set.

#### `-thresh percentage`

Specifies a percentage of the storage pool's estimated capacity that, when reached or exceeded, causes the metadata server to generate an alert. If the capacity drops and then reaches the percentage again, additional alerts are sent.

You can specify a value between 0 and 100. The default alert threshold percentage is 80. If set to 0, no alerts are generated.

#### `-desc description`

Specifies a description for the storage pool. The description must be enclosed

in matching single (') or double (") quotation marks if it contains any blank characters. This description can be up to 256 characters in length.

The default is an empty string.

#### *pool\_name*

Specifies the names of one or more storage pools to create. Each name can be up to 256 characters in length.

- Specifies that you want to read the names of one or more storage pools to create from stdin (for example, – << /work/stgpools\_list.txt).

### Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

### Description

You can create multiple storage pools with the same values for the partition size, allocation size, and alert threshold in a single command; however, you cannot set the name or description for multiple storage pools.

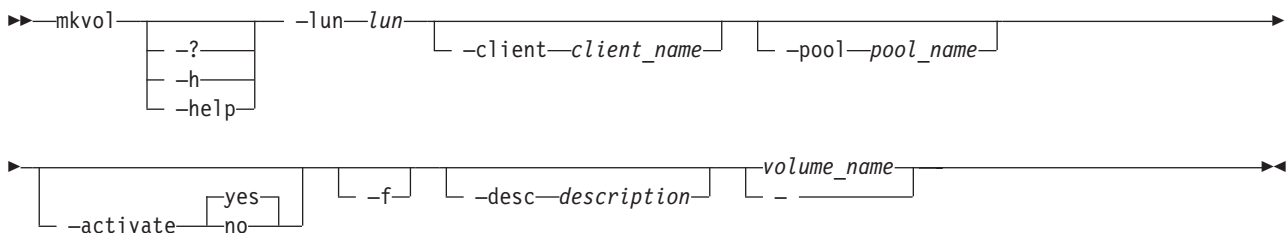
### Example

**Create two storage pools** The following example creates two storage pools (stgpool1 and stgpool2), sets the partition size on both to 256 MB, and specifies to send alerts when the allocated capacity reaches 70%.

```
sfscli> mkpool -partsize 256 -thresh 70 stgpool1 stgpool2
Storage pool stgpool1 created.
Storage pool stgpool2 created.
```

## mkvol

Adds one or more volumes to a storage pool.



### Parameters

#### **-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### **-lun lun**

Specifies the identifier of a logical unit (LU) to make into a volume. The identifier is comprised of the logical unit number (LUN) format, an equal sign, and the LUN ID.

**Example:** vpd83naa5=12234678

**-client** *client\_name*

Specifies the name of a client that is zoned to access the LU.

To create a volume in a user storage pool, you must specify a SAN File System client that meets the following conditions:

- The client must have access to the LU that is being added as a SAN File System volume. To list all active clients that can access the LU, use the **reportclient -lun** command.
- The client must be active and appear in the registered (active) client list.

Do not specify a client if you are creating a volume in the system storage pool.

**-pool** *pool\_name*

Specifies the name of the storage pool to which to add the new volumes. The storage pool is either a user-defined, default, or system storage pool. If not specified, this command adds the volume to the default storage pool.

**-activate**

Specifies whether the volume is activated. The metadata server can allocate new data only on activated volumes. Possible values are the following:

- |            |                                                                                                    |
|------------|----------------------------------------------------------------------------------------------------|
| <b>yes</b> | Specifies that the metadata server can allocate new data on the volume. This is the default value. |
| <b>no</b>  | Specifies that the metadata server cannot allocate new data on the volume.                         |

- f** Forces the metadata server to add the volume and write a new label to the volume if the volume already has a valid SAN File System label.

**Tip:** You can use **-f** only if the volume is not assigned to another storage pool in the same cluster.

**-desc** *description*

Specifies the description for the volume. The description must be enclosed in matching single (') or double (") quotation marks if it contains any blank characters. This description can be up to 256 characters in length.

The default is an empty string. Specifying this parameter with no description results in a syntax error.

**Note:** Do not specify this parameter for volumes that you are adding to the system storage pool.

*volume\_name*

Specifies the names of one or more volumes to add. This name must be unique within the cluster (not necessarily the fileset) and can be up to 256 characters in length.

- Specifies that you want to read the names of one or more volumes to add from stdin (for example, - << /work/vol\_list.txt).

## Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

You must format the applicable LUNs before invoking this command.

The cluster must not be in an offline state when you run this command.

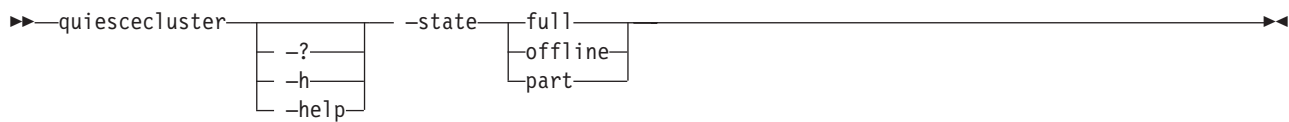
### Example

**Create a volume** The following example creates a volume and adds it to the storage pool named **fast**:

```
sfsccli> mkvol -lun vpd83naa5=12234678 -client client03 -pool fast -desc "fast
vol 1" first_vol
Volume first_vol successfully created.
```

## quiescecluster

Changes the state of all metadata servers in the cluster to one of three quiescent states.



### Parameters

#### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### -state

Specifies the state to which you want to change the cluster. You can specify one of these states:

**full** Changes the state to fully quiescent. This parameter flushes all client caches and long-running processes, resulting in no activity on the metadata servers.

You would put the cluster in this state to produce self-consistent LUN-based backups for a known FlashCopy image.

#### offline

Changes the state to offline. This parameter terminates all current client sessions and prevents new client sessions from being started, but allows all metadata server I/O to continue. The offline state is intended to be used as an administrative mode to, for example, reassign a fileset to another metadata server.

You would put the cluster in this state to restrict clients from accessing the cluster.

#### part

Changes the state to partly quiescent. This parameter prevents new client sessions from starting. Open files are still accessible, but not new files can be opened or created while in this state.

You would put the cluster in this state to produce *dirty* LUN-based backups without requiring clients and applications to stop, and to allow client applications that use direct I/O (such as database servers) to back up their data.

## Prerequisites

You must have Operator or Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

The quiescent states restrict activity on all metadata servers in the cluster. You would put the metadata servers in this state in preparation for backup tasks that require the client sessions, file I/O, client metadata, or system metadata to be inactive.

You can use the **resumecluster** command to return all the metadata servers in the cluster to the online state.

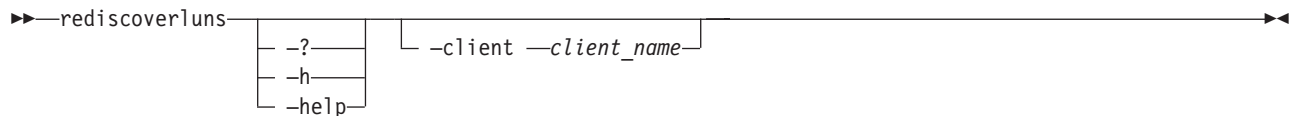
## Example

**Change the cluster state** The following example changes the cluster state to fully quiescent:

```
sfscli> quiescecluster -state full
Cluster successfully in fully quiescent state.
```

## rediscoverluns

Rediscover logical unit numbers (LUNs) on the SAN that are visible to SAN File System. The metadata LUNs are rediscovered if no parameters are specified.



## Parameters

### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### -client *client\_name*

Rediscover the LUN information cached by specified client.

## Description

This command operates only against the local engine.

All of the metadata servers can see the same LUNs.

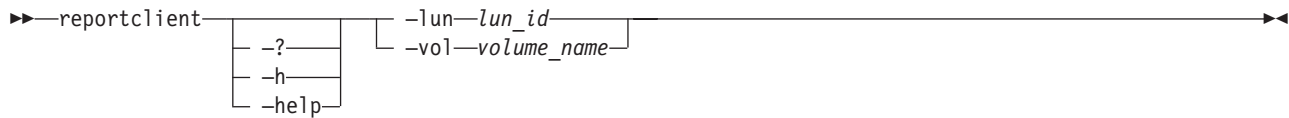
## Example

**Rediscover LUNs** The following example rediscover LUN information that is cached by client client03.

```
sfscli> rediscoverluns -client client03
The LUNs have been rediscovered. Tip: Run lsln to view the LUNs.
```

## reportclient

Displays a list of clients that have access to the specified volume or logical unit number (LUN).



### Parameters

#### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### -lun *lun\_id*

Specifies the LUN ID for which you want to display a list of clients that can access the LUN. Clients that are zoned with the same permissions have access to the same LUNs. If the LUN contains spaces, you must enclose the ID in single quotation marks (') or double quotation marks (").

#### -vol *volume\_name*

Specifies the volume name for which you want to display a list of clients that can access it. Clients that are zoned with the same permissions have access to the same LUNs.

### Prerequisites

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

### Description

This command displays a list of names that can access the specified LUN or volume.

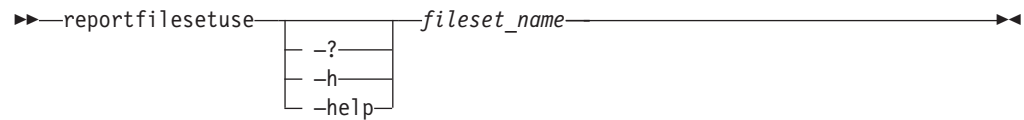
### Example

**Stop the metadata check process** The following example lists all clients that have access to volume Vol\_1.

```
sfscli> reportclient -vol Vol_1
Name
=====
client01
client02
client03.
```

## reportfilesetuse

Displays the usage statistics for pools that currently store data for the specified fileset.



## Parameters

### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

*fileset\_name*

Specifies the name of the fileset to report.

## Prerequisites

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

This command displays the following information for each storage pool:

- Storage pool name.
- Fileset usage (In Use or Not In Use).
- Number of rules in the active policy that place files into the storage pool.

## Example

**List fileset usage** The following example displays usage statistics for fileset cnt\_2:

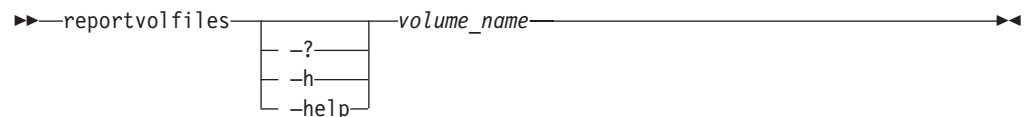
```

sfsccli> reportfilesetuse cnt_2
Pool Fileset Usage Total Rules That Enable Usage
=====
pool_1 In Use 0
pool_2 Not In Use 4
pool_3 In Use 7

```

# reportvolfiles

Displays a list of files, and their attributes, in the specified volume.



## Parameters

### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

*volume\_name*

Specifies the name of the volume to report.



## Prerequisites

You must have Backup, Operator, or Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

This command displays the following information for each file in the specified volume:

- Fileset name
- Fileset directory
- Relative file path
- File name

This information is displayed in the following format:

*fileset:fileset\_directory/relative\_file\_path/file\_name*

This command is useful for performing a file-based restore operation after a user volume failure because it can list the files in a failed user volume; however, it cannot list files on a failed system volume.

**Tip:** This command gathers all the data before displaying any output. If there are a very large number of files on the volume, there could be a delay of several minutes before output is displayed.

## Example

**List files in a volume** The following example displays a list of files in volume vol2:

```
sfscli> reportvolfiles vol2
Root:Work/doc/language.doc
NotesR5D:InfoDev/IDPlan.doc
Resources:Personnel/dept/manager/people.txt
```

## resetadmuser

Forces all administrative users to log in again.



## Parameters

`-? | -h | -help`

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

## Prerequisites

You must have Administrator privileges to use the command.

## Description

**Note:** This command operates only against the local engine.

To increase Common Information Model (CIM) performance, the administrative agent retains a copy of the administrative user's password after the LDAP server validates it. This copy is used for a certain period of time to avoid the overhead of going to the LDAP server to authenticate the user for every CIM request. If a user's password has been changed in the LDAP server, use this command to ensure that the user must immediately use the new password rather than continuing to use the administrative agent's copy.

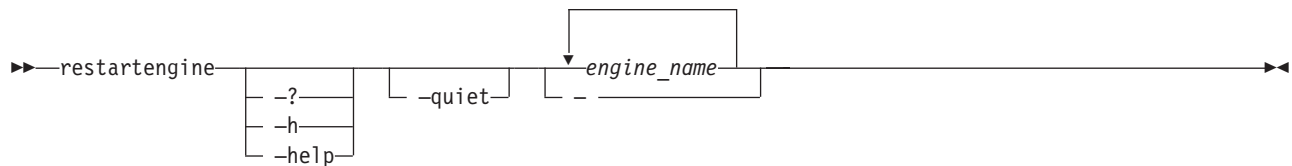
## Example

**Reset the Administrative Users' Passwords** The following example resets all administrative user's passwords in the CIM cache:

```
sfsccli> resetadmuser
Users reset.
```

## restartengine

Reboots one or more storage engines.



## Parameters

### `-? | -h | -help`

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### `-quiet`

Turns off confirmation messages for this command.

### `engine_name`

Specifies the names of one or more storage engines to restart.

- Specifies that you want to read the names of one or more storage engines to restart from stdin (for example, `- << /work/engine_list.txt`).

## Prerequisites

You must have Operator or Administrator privileges to use the command.

## Description

You can stop and restart the local engine, but you cannot start the local engine.

## Example

**Restart engines** The following example restarts engine ST1:

```
sfsccli> restartengine ST1
Are you sure you want to restart engine ST1? [y/n] Y
Engine ST1 restarted successfully.
```

## resumecluster

Brings all metadata servers in the cluster to the online state.

```

>> resumecluster
 -?
 -h
 -help

```

### Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### Prerequisites

You must have Operator or Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

### Example

**Resume the cluster** The following example resumes normal activity on the cluster:

```
sfsccli> resumecluster
Cluster successfully returned to online state.
```

## reverttoimage

Reverts the current fileset to a specified FlashCopy image of the file layout and contents.

```

>> reverttoimage
 -?
 -h
 -help
 -quiet
 -fileset fileset_name
 -f
 image_name

```

### Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

**-quiet**

Turns off confirmation messages for this command.

**-fileset** *fileset\_name*

Specifies the name of the fileset that you want to revert to a previous FlashCopy image.

**-f** Forces the metadata server to revert to the specified FlashCopy image if a

client is reading files in the current image. Changes made to files in the fileset since the specified FlashCopy image was created will be lost.

*image\_name*

Specifies the name of the FlashCopy images to which you want to revert.

- Specifies that you want to read the name of the FlashCopy image to which you want to revert from stdin (for example, – << /work/image\_list.txt).

### Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

### Description

**Attention:** When you revert a fileset to a specified target FlashCopy image, the target FlashCopy image and all FlashCopy images taken between the current fileset and target FlashCopy image are deleted. The target FlashCopy image becomes the primary image for the fileset and no longer appears as an image listed in the .flashcopy directory.

Unless you specify the **-f** parameter, there *must not* be any client activity in the FlashCopy images being deleted or in the current fileset image.

You cannot revert to a FlashCopy image when nested filesets exist within the fileset. You must manually detach the nested filesets before running the **reverttoimage** command.

The .flashcopy directory is a hidden directory. Windows-based clients must use special operating-system commands to view this directory.

Depending on the age of the specified FlashCopy image and the amount of unique file data in the image tree, the revert operation could result in significant background activity to clean up the file-system objects that are no longer referenced.

Until the **reverttoimage** command completes, you cannot perform any activity on the fileset except the **rmimage** and **reverttoimage** commands. If you issue a second **reverttoimage** command before the first one completes, you can only revert to a FlashCopy image that existed before the FlashCopy image being reverted by the first command.

Because the specified FlashCopy image is deleted after you issue the **reverttoimage** command, it is recommended that you preserve a secondary backup of the fileset before using the command for future use or disaster recovery.

To preserve the fileset hierarchy, you must reattach any nested filesets that you detached before issuing the command when this command completes. It is recommended that you note the hierarchy using the **lsfileset** command before reverting to a FlashCopy image.

## Example

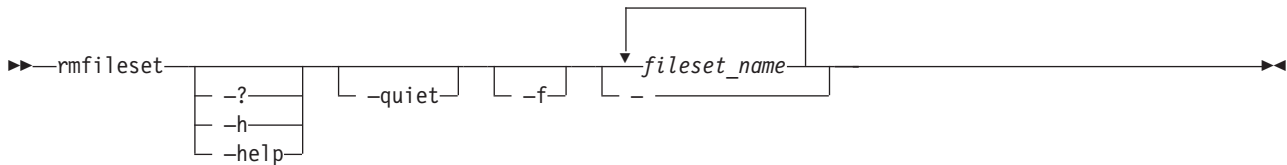
**Revert to a previous FlashCopy image** The following example reverts the fileset *cnt\_A* to the FlashCopy image *cntA\_image*:

```
sfscli> reverttoimage -fileset cnt_A cntA_image
```

```
Are you sure you want to revert to FlashCopy image cntA_image? y/n Y
Image "cntA_image" successfully restored.
```

## rmfileset

Removes one or more empty, detached filesets and optionally the files in the filesets, including any FlashCopy images.



### Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

**-quiet**

Turns off confirmation messages for this command.

**-f** Forces the metadata server to delete any files and directories before removing the filesets.

*fileset\_name*

Specifies the names of one or more filesets to remove.

**-** Specifies that you want to read the names of one or more filesets to remove from stdin (for example, `- << /work/cnt_list.txt`).

### Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

### Description

Unless you specify the `-f` parameter, the fileset *must* be detached and empty before it can be deleted.

You cannot remove the ROOT fileset or a fileset that is reference by the active policy.

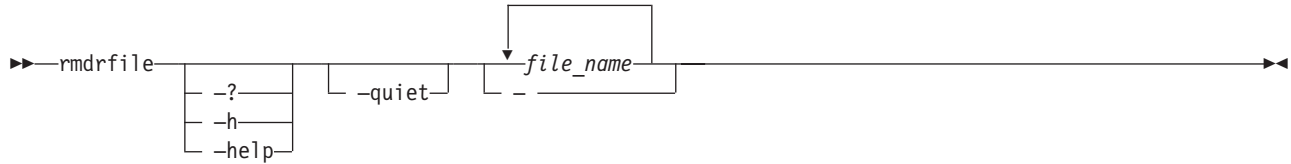
## Example

**Remove a fileset** The following example removes the fileset named *cnt\_A*:

```
sfsccli> rmfileset cnt_A
Are you sure you want to delete Fileset cnt_A? [y/n] Y
Fileset cnt_a removed.
```

## rmldrfile

Deletes an existing system-metadata disaster-recovery dump file.



### Parameters

#### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### -quiet

Turns off confirmation messages for this command.

#### *file\_name*

Specifies the names of one or more system-metadata disaster-recovery dump files to remove.

- Specifies that you want to read the names of one or more system-metadata disaster-recovery dump files to remove from stdin.

### Prerequisites

You must have Backup, Operator, or Administrator privileges to use the command.

### Description

**Note:** This command operates only against the local engine.

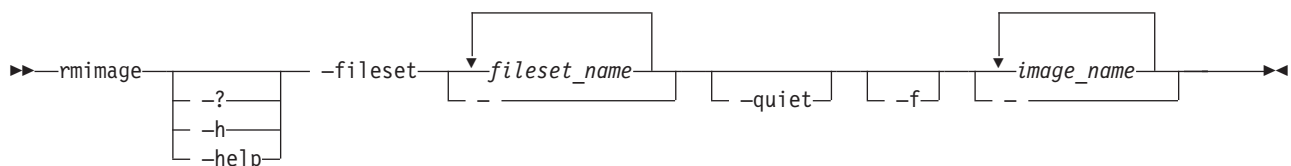
### Example

**Remove a dump file** The following example removes the dump file named dr1:

```
sfsccli> rmldrfile dr1
Disaster recovery file "dr1" removed.
```

## rmimage

Deletes one or more FlashCopy images for a specific fileset.



## Parameters

### **-?** | **-h** | **-help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### **-fileset** *fileset\_name* | **-**

Specifies the names of one or more filesets from which to delete one or more FlashCopy images. Specifying a **-** indicates that you want to read the names of one or more filesets from stdin (for example, **-** << /work/cnt\_list.txt).

### **-quiet**

Turns off confirmation messages for this command.

### **-f**

Forces the metadata server to delete a FlashCopy image if one or more files are open in the image tree. Changes that were made to the open files but not saved will be lost.

### *image\_name*

Specifies the names of one or more FlashCopy images to delete.

### **-**

Specifies that you want to read the names of one or more FlashCopy images to delete from stdin (for example, **-** << /work/image\_list.txt).

## Prerequisites

You must have Backup, Operator, or Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

Unless you specify the **-f** parameter, you cannot delete a FlashCopy image if there is client activity on that image.

Depending on the age of the FlashCopy image and the amount of unique file data in the image tree, the delete operation might result in significant background activity to clean up the file-system objects that are no longer referenced.

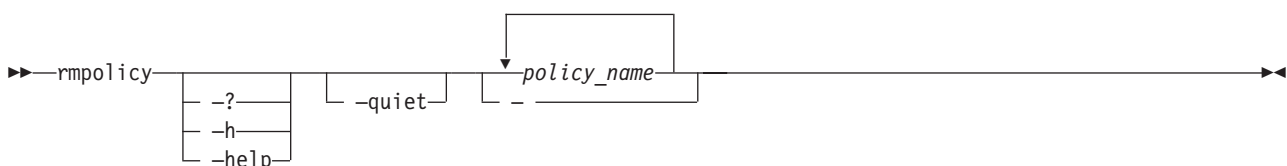
## Example

**Delete a FlashCopy image** The following example deletes the FlashCopy image (*CA\_image*) from the fileset *cnt\_A*:

```
sfscli> rmimage -fileset cnt_A CA_image
Are you sure you want to delete FlashCopy image CA_image? [y/n] Y
Image "CA_image" successfully deleted.
```

## rmpolicy

Deletes one or more inactive policies.



## Parameters

### **-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### **-quiet**

Turns off confirmation messages for this command.

### *policy\_name*

Specifies the names of one or more inactive policies to delete.

- Specifies that you want to read the names of one or more inactive policies to delete from stdin (for example, - << /work/policies\_list.txt).

## Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the the operating system on engine hosting the master metadata server to run this command.

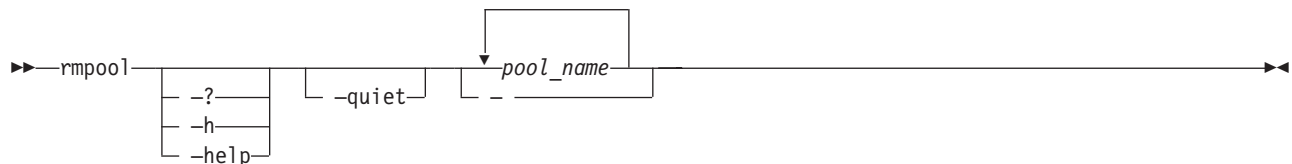
## Example

**Remove a policy** The following example removes the policy named *test\_policy*:

```
sfscli> rmpolicy test_policy
Are you sure you want to delete policy test_policy? [y/n] Y
Policy test_policy removed.
```

# rmpool

Deletes one or more empty, unreferenced storage pools.



## Parameters

### **-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### **-quiet**

Turns off confirmation messages for this command.

### *pool\_name*

Specifies the names of one or more storage pools to delete.

- Specifies that you want to read the names of one or more storage pools to delete from stdin (for example, - << /work/stgpools\_list.txt).

## Prerequisites

You must have Administrator privileges to use the command.



You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

### Description

You cannot delete a storage pool that is not empty. You must remove all volumes from the storage pool before you can delete it.

You cannot delete a storage pool that is referenced by the active policy.

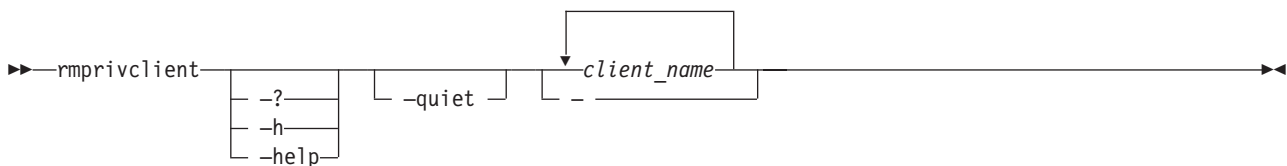
### Example

**Remove storage pools** The following example removes storage pool stgpool1.:

```
sfscli> rmpool stgpool1
Are you sure you want to delete storage pool stgpool1?
Storage pool stgpool1 removed.
```

## rmprivclient

Revokes privileged access to the SAN File System global namespace from the specified clients.



### Parameters

#### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### -quiet

Turns off confirmation messages for this command

#### client\_name

Specifies one or more names of clients whose root user (on UNIX) or Administrator user (on Windows) is to be revoked privileged access to the SAN File System global namespace.

- Specifies that you want to read the names of one or more clients from stdin (for example, - << /work/client\_list.txt).

### Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

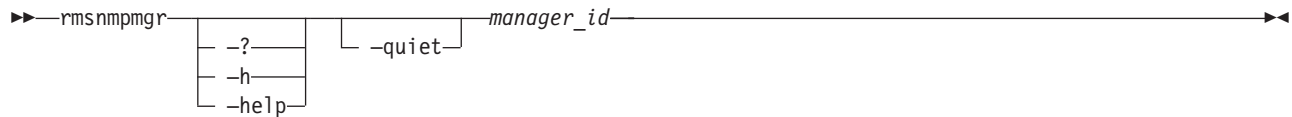
### Example

**Revoke root privileges** This example revokes root privileges to the client named client03:

```
sfsccli> rmprivclient client03
Are you sure you want to remove client03 as a privileged client? [y/n] Y
Privilege client access successfully removed for client03.
```

## rmsnmpmgr

Removes an SNMP manager (recipient).



### Parameters

#### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### -quiet

Turns off confirmation messages for this command.

#### manager\_id

Specifies the number that identifies the SNMP manager (recipient) to delete. Because this number is not static, you must acquire the new number each time a change is made to the list of managers.

### Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

### Description

If all SNMP managers are removed, no SNMP traps are generated.

When you change a disruptive cluster setting, dynamic cluster settings (such as SNMP settings) cannot be modified until you reboot the cluster.

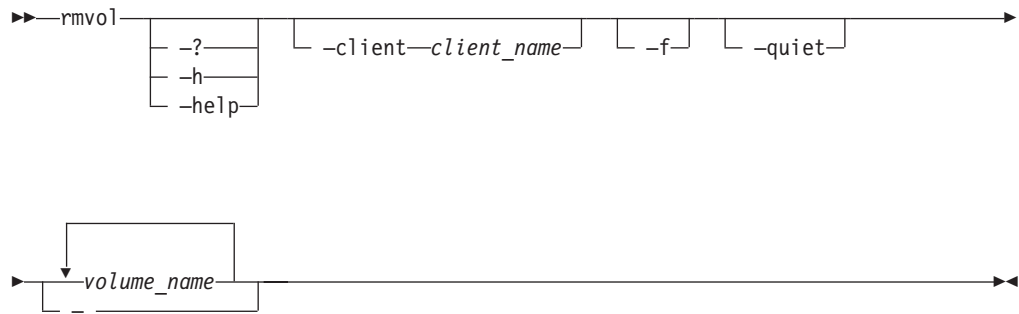
### Example

**Removes an SNMP manager** The following example removes an SNMP manager:

```
sfsccli> rmsnmpmgr 1
Are you sure you want to remove SNMP manager 1? [y/n] Y
SNMP manager 1 successfully removed.
```

## rmvol

Removes one or more volumes from a storage pool and redistributes the contents to other volumes in the same storage pool.



## Parameters

### **-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### **-client** *client\_name*

Specifies the name of a client that is zoned to access the LU.

To remove a volume from a user storage pool, you must specify a SAN File System client that meets the following conditions:

- The client must have access to the LU that is being removed as a SAN File System volume. To list all active clients that can access the volume, use the **reportclient -vol** command.
- The client must be active and appear in the registered (active) client list.
- The client must be able to access all volumes in the storage pool from which the volume is being removed. To verify that the client can see all volumes in the storage pool, use the **reportclient -lun** command, and then use the **lsvol -pool** command to view the volumes in the storage pool and cross check the results.

**Note:** Do not specify a client if you are removing a volume from the system storage pool.

- f** Removes all files in the specified volumes and disables draining the files to another volume in the storage pool. If there is a file in the volume, specifying this parameter causes the entire file to be deleted, even if parts of the file reside on multiple volumes.

The **-f** parameter is not allowed when you remove volumes from the system storage pool.

**Note:** There is no automatic-recovery process when you specify this parameter. You can look at the activity log to get a list of the files for which failures occurred, and then restore those files manually. You can also use the **reportvolfiles** command before invoking the **rmvol** command and review the file list to make sure that the files are not needed or are archived.

### **-quiet**

Turns off confirmation messages for this command.

### *volume\_name*

Specifies the names of one or more volumes to delete.

- Specifies that you want to read the names of one or more volumes to delete from stdin (for example, `- << /work/vol_list.txt`).

### Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

The cluster must not be in an offline state when you run this command.

### Description

You do not need to deactivate a volume before you remove it.

Before removing a volume, SAN File System moves (drains) the contents of the volume across other available volumes in the same storage pool. If the storage pool does not have sufficient space available in other volumes to move all of the data contained in the specified volume, the removal fails and the metadata server suspends the volume (the metadata server cannot allocate new data on that volume).

A volume must be empty to be removed from a storage pool. If one or more files cannot be moved to another volume in the same storage pool, you can specify the `-f` parameter to discard those files. If you do not specify the `-f` parameter and the files cannot be removed, the removal fails.

If an error occurs during volume removal, the volume remains in a suspended state so new allocations will not be satisfied from the volume being removed. To activate the volume, use the **activatevol** command.

You must explicitly remove a volume if you want to reuse it. Removing a volume removes the label and frees the device for reuse. If you want to add a device that is in the in-use state, you must remove it using the `-f` option before you can add it as a volume.

### Example

**Remove two volumes** The following example removes volumes vol1 and vol2:

```
sfscli> rmvol -client client02 vol1 vol2
Are you sure you want to delete volume vol1? [y/n] Y
Volume vol1 removed.
Are you sure you want to delete volume vol2? [y/n] Y
Volume vol2 removed.
```

## setautorestart

Modifies the metadata server restart service settings.

```
►►—setautorestart—┐ —quiet —┐ ┐ —probeinterval —seconds—┐ ┐ —livenesstimeout —seconds—┐
 └──┘
►┐ —retrylimit —count—┐ ┐ —restarttimeout —seconds—┐
└──┘◄◄
```

## Parameters

### **-quiet**

Turns off confirmation messages for this command.

### **-probeinterval** *seconds*

Specifies the time, in seconds, between metadata server restart service probes. You can specify a value from 10 to 60. The initial value is 10.

### **-livenesstimeout** *seconds*

Specifies the time, in seconds, to wait for the metadata server to respond to a metadata server restart service probe. You can specify a value from 1 to 10. The initial value is 1.

### **-retrylimit** *count*

Specifies number of times the metadata server restart service probe can fail before considering that metadata server down. You can specify a value from 1 to 10. The initial value is 3.

If the retry limit is exceeded, the metadata server restart service is disabled.

### **-restarttimeout** *seconds*

Specifies the amount of time, in seconds, that the administrative agent will wait for the metadata server to restart before returning a timed-out error and disabling the metadata server restart service. For example, the master metadata server must become operational, before the subordinate metadata servers can be started.

You can specify a value from 1 to 600. The initial value is 600 seconds. If the restart timeout is exceeded, the metadata server restart service is disabled.

**Tip:** This value should typically be set high to avoid false failures.

## Prerequisites

This task must be performed only by trained service personnel.

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

**Attention:** You must reboot the metadata server, if you modified any of these settings, before changes can be applied.

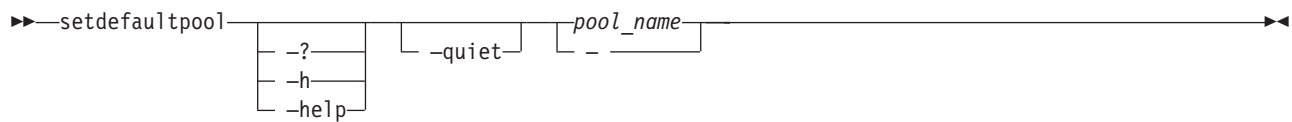
## Example

**Modify metadata server restart service settings** The following example modifies the probe interval setting for metadata server ST0:

```
sfscli> setautorestart -probeinterval 60 ST0
Are you sure you want to change the automatic restart service configuration settings
[y/n] Y
The automatic restart service configuration was successfully modified on all
of the servers in the cluster.
```

## setdefaultpool

Changes a user storage pool to the default storage pool, and changes the previous default storage pool to a regular, nondefault user storage pool.



## Parameters

### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### -quiet

Turns off confirmation messages for this command.

### pool\_name

Specifies the name of the storage pool to set as the default.

- Specifies that you want to read the name of the storage pool to set as the default from stdin (for example, - << /work/stgpool\_list.txt).

## Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

You cannot change the system storage pool using this command.

To disable the use of the default storage pool and make the current default storage pool a regular user storage pool, you can use the **disabledefaultpool** command.

## Example

**Set the default storage pool** The following example changes the storage pool named **default\_stgpool** to be the new default storage pool:

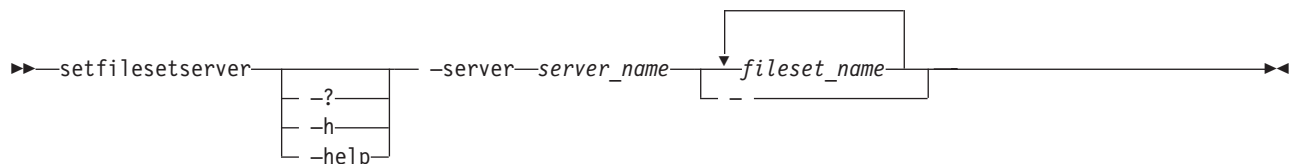
```
sfscli> setdefaultpool default_stgpool
```

```
Are you sure you want to set default_stgpool as the default pool? [y/n] Y
```

```
Pool default_stgpool is now the DEFAULT pool.
```

## setfilesetserver

Reassigns an existing fileset to be hosted by a different metadata server.



## Parameters

### **-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### **-server** *server\_name*

Specifies the new metadata server to host the fileset.

If the specified metadata server goes down, the fileset is moved to another metadata server temporarily. When the assigned metadata server comes back online, the fileset is moved back to its assigned metadata server.

### *fileset\_name*

Specifies the names of one or more filesets to reassign to a new metadata server.

- Specifies that you want to read the names of one or more filesets to reassign from stdin (for example, - << /work/cnt\_list.txt).

## Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

The current metadata server must be in one of the following states:

- Not running
- Offline
- Online

The target metadata server must be part of the cluster and must be in one of the following states:

- Not running
- Offline
- Online

## Description

If current metadata server is offline, and the target metadata server is either online or offline, then the state of the cluster does not change.

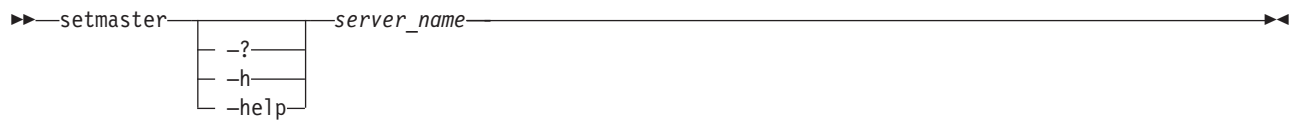
## Example

**Reassign a fileset** The following example reassigns two filesets (**cnt\_A** and **cnt\_B**) to metadata server ST1:

```
sfsccli> setfilesetserver -server ST1 cnt_A cnt_B
Fileset cnt_A assigned to server ST1.
Fileset cnt_B assigned to server ST1.
```

## setmaster

Sets a subordinate metadata server as the new master metadata server in the event of an irrecoverable loss of the current master metadata server.



## Parameters

### **-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### *server\_name*

Specifies the name of the subordinate metadata server to set as the master metadata server.

## Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the *subordinate* metadata server to run this command.

## Description

**Attention:** This command is not available in SAN File System version 2.1. It is available only during a rolling upgrade from SAN File System version 1.1 to version 2.1.

**Attention:** When you invoke the **setmaster** command, applications will notice a pause in service, which could adversely affect application that are sensitive to time outs.



**Attention:** If the master metadata server is not shut down, you could destroy all of your data. Before using the `setmaster` command, perform the following steps to verify that the master metadata server is not shut down:

1. From the engine hosting the master metadata server, perform the following steps:
  - a. Verify that the master metadata server is *offline* by invoking the **lsserver** command.
  - b. If the master metadata server is *online*, invoke the **stopserver** command to stop the master metadata server.  
Do not stop the subordinate metadata servers.
  - c. Verify again that the master metadata server is *offline* by invoking the **lsserver** and **statcluster -netconfig** commands.
2. Verify that all other metadata servers are subordinates by invoking the **lsserver** command from each engine hosting the remaining metadata servers.
3. From the engine hosting the subordinate metadata server that you want to become the master, perform the following steps:
  - a. Verify that the engine hosting the master metadata server is shut down by invoking the **lsengine** command.  
If the engine is running, use the **stopengine** or **stopengine -f** command to shut it down.  
  
**Note:** If you cannot shut down the engine remotely or through the Administrative CLI, manually power off the engine.
  - b. Verify again that the engine hosting the master metadata server is shut down by invoking the **lsengine** command.

Before stopping the master metadata server, you should move the filesets that are being served by the master metadata server to another metadata server.

Setting a new master metadata server may affect the clients. If the clients mount to the global namespace using the IP address of the master metadata server, this address will become unavailable if the client reboots. Therefore, you must reset the IP address to which each client mounts. For AIX client, edit the `/usr/tank/client/config/stclient.conf` file, and set the `server_hot_name` variable.

The **setmaster** command demotes the master metadata server to a subordinate. It does not drop the metadata server from the cluster. When you restart the engine, the metadata server joins the cluster as subordinate.

This command is useful when you want to replace hardware components of or upgrade the engine hosting the master metadata server.

### Example

**Set the master metadata server** The following example verifies that the master metadata server ST0 is shut down and then sets the subordinate metadata server ST1 as the master metadata server.

From metadata server ST0:

```
sfscli> lsserver
Name State Server Role Filesets Last Boot
=====
ST0 Online Master 0 Jul 10, 2003 7:00:17 AM
ST1 Online Subordinate 2 Jul 10, 2003 7:00:24 AM
```

```
sfscli> stopservre ST0
Server "ST0" stopped gracefully.
```

```
sfscli> lsserver
Name State Server Role Filesets Last Boot
=====
ST0 Not Running Master - -
ST1 Joining Subordinate 2 Jul 10, 2003 7:00:24 AM
```

```
sfscli> statcluster -netconfig
Could not connect to the server. Please start the server.
```

From metadata server ST1:

```
sfscli> lsserver
Name State Server Role Filesets Last Boot
=====
ST1 Joining Subordinate 2 Jul 10, 2003 7:00:24 AM
```

```
sfscli> statcluster -netconfig
statcluster must be issued from the admin master server.
```

```
sfscli> stopengine ST0
Are you sure you want to stop engine "ST0"? [y/n] Y
Engine "ST0" was shutdown successfully.
```

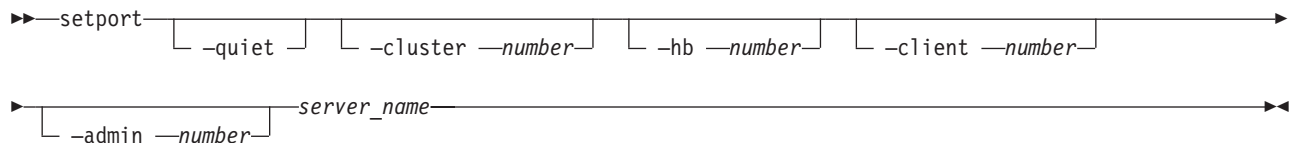
```
sfscli> lsengine
Name Engine IP Boot State Temp Fans Voltage
=====
ST1 555.168.10.25 In OS Normal Normal Normal
ST0 555.168.10.24 Unknown/Power Off - - -
```

```
sfscli> setmaster ST1
Server "ST1" is the new master server.
```

```
sfscli> lsserver
Name State Server Role Filesets Last Boot
=====
ST1 Online Master 2 Jul 10, 2003 7:05:12 AM
ST0 Unknown Subordinate 2 Jul 10, 2003 7:05:12 AM
```

## setport

Sets the port values for the metadata server.



### Parameters

#### -quiet

Turns off confirmation messages for this command

#### -cluster *number*

Specifies a new cluster port for the metadata server. The initial cluster port is 1737.

#### -hb *number*

Specifies a new heartbeat port for the metadata server. The initial heartbeat port is 1738.

**-client** *number*

Specifies a new client-server port for the metadata server. The initial client-server port is 1700.

**-admin** *number*

Specifies a new administrative port for the metadata server. The initial administrative port is 1800.

*server\_name*

Specifies the name of the metadata server for which to set the port values.

## Prerequisites

This task must be performed only by trained service technicians.

You must have Administrator privileges to use the command.

## Description

If you run this command from an engine hosting a subordinate metadata server, you can set the port only for the local metadata server. If you run this command from an engine hosting the master metadata server, you can set the port for any metadata server.

**Requirement:** You must reboot the metadata server, if you modified any of these settings, before changes can be applied.

Setting incorrect port values might cause the metadata server to cease functioning.

When the administrative agent starts and notices that the tank.properties file has overriding parameters in it, the administrative agent reads these values and uses them.

## Example

**Set the cluster port** The following example sets the cluster port number on metadata server ST0:

```
sfscli> setport -cluster 5669 ST0
```

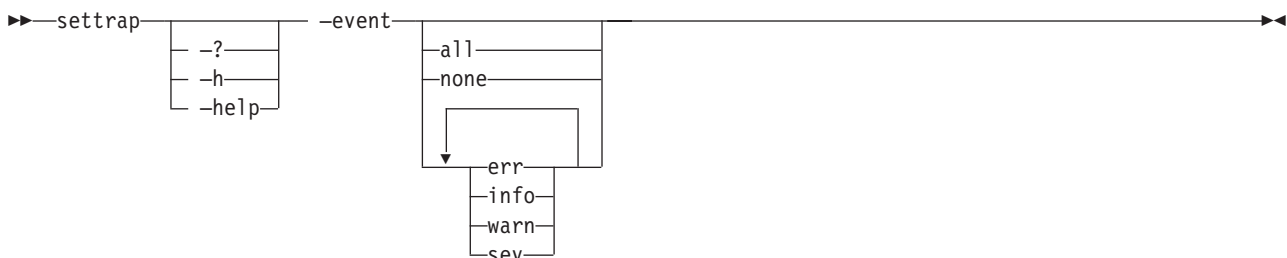
```
Are you sure you want to change the server port value, which will require a
server restart for changes to apply? [y/n] Y
```

```
Cluster port settings on Metadata server ST0 successfully modified.
```

```
Restart the server for all changes to apply.
```

## settrap

Specifies whether an SNMP trap is generated and sent to all SNMP managers when a specific type of event occurs on the metadata server.



## Parameters

### **-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### **-event**

Specifies the events for which traps are to be sent. You can specify any of the following values:

- all** Generates an SNMP trap when an information, error, warning, or severe event occurs. This parameter cannot be combined with other values.
- none** Turns off SNMP traps for all events. This value cannot be combined with other values.
- info** Generates an SNMP trap when an information event occurs. This value can be combined with any value except all or none. Multiple values must be separated by a comma but no space.
- err** Generates an SNMP trap when an error event occurs. This value can be combined with any value except all or none. Multiple values must be separated by a comma but no space.
- warn** Generates an SNMP trap when a warning event occurs. This value can be combined with any value except all or none. Multiple values must be separated by a comma but no space.
- sev** Generates an SNMP trap when a severe event occurs. This value can be combined with any value except all or none. Multiple values must be separated by a comma but no space.

## Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

**Note:** To generate traps, you must set the SNMP event types that you want to generate traps and you must add an SNMP manager (using the **addsnmpmgr** command).

SNMP traps of the specified type are generated for all SNMP managers.

When you change a disruptive cluster setting, dynamic cluster settings (such as SNMP settings) cannot be modified until you reboot the cluster.

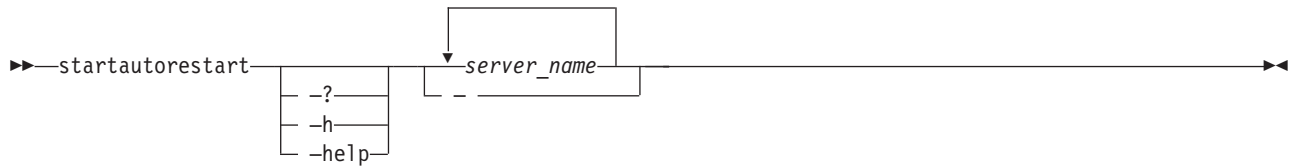
## Example

**Sets the SNMP traps** The following example sets SNMP traps for severe and warning events:

```
sfscli> settrap -event sev,warn
SNMP trap event level was successfully set.
```

## startautorestart

Enables the metadata server to restart automatically if it is down.



### Parameters

#### `-?` | `-h` | `-help`

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### *server\_name*

Specifies the names of one or more metadata servers to enable them to restart automatically.

- Specifies that you want to read the names of one or more metadata servers to enable to restart automatically from stdin (for example, `- << /work/server_list.txt`).

### Prerequisites

You must have Administrator privileges to use the command.

### Description

**Note:** If you run this command from an engine hosting a subordinate metadata server, you can enable the metadata server restart service on only the local metadata server. If you run this command from the engine hosting the master metadata server, you can enable the metadata server restart service on any metadata server.

The metadata server restart service is enabled by default.

When a metadata server is enabled to restart automatically, an SNMP trap is not sent when the metadata server is restarted.

Manually stopping a metadata server or cluster disables the metadata server restart service for that metadata server or cluster. Manually starting the metadata server or cluster reenables the metadata server restart service for that metadata server or cluster.

If the metadata server restart service attempts to restart a metadata server and fails, the metadata server restart service tries again to restart the metadata server, up to the retry limit. After the retry limit, the metadata server restart service is disabled.

### Example

**Enable the metadata server restart service** The following example enables the metadata server restart service for metadata server ST1:

```
sfsccli> startautorestart ST1
The automatic restart service for server ST1 successfully enabled.
```

## startcluster

Starts all metadata servers in the cluster and brings them to the full online or offline state.



### Parameters

#### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### -offline

Starts all metadata servers in the cluster in the offline state. This option is useful in disaster recovery situations.

### Prerequisites

You must have Operator or Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

### Description

None of the metadata servers in the cluster can be running when using this command.

When you start all of the metadata servers in the cluster using the **startcluster** command, all of the subordinate metadata servers are brought to the online state. When you start the metadata server using the **startserver** command, the subordinate metadata servers in the joining state are placed in their previous state after the master metadata server is up and running.

**Tip:** A message stating that the cluster started successfully does not necessarily mean that the cluster is online.

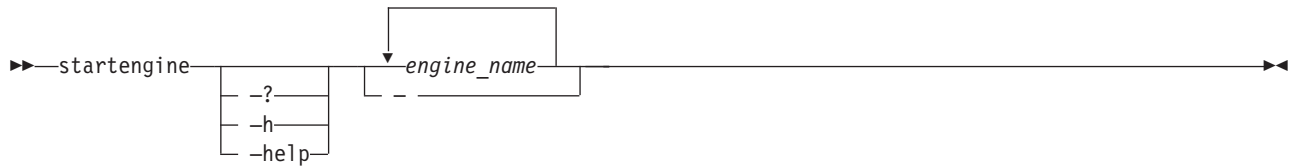
### Example

**Start the cluster** The following example starts all metadata servers in the cluster in the offline state:

```
sfsccli> startcluster -offline
Cluster started successfully in the offline state.
```

## startengine

Starts the CPU and operating system on one or more storage engines.



### Parameters

#### `-?` | `-h` | `-help`

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### `engine_name`

Specifies the name of one or more storage engines to start.

- Specifies that you want to read the names of one or more storage engines to start from stdin (for example, `- << /work/engine_list.txt`).

### Prerequisites

You must have Operator or Administrator privileges to use the command.

### Description

You can stop and restart the local engine, but you cannot start the local engine.

### Example

**Start an engine** The following example starts engine ST1:

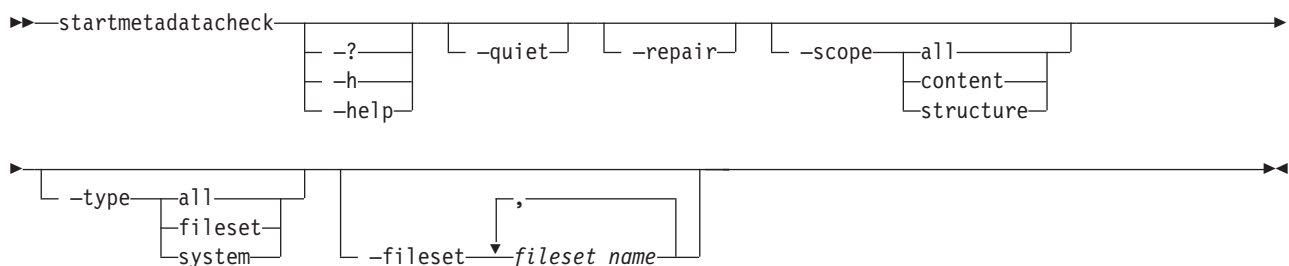
```
sfscli> startengine ST1
```

A start request has been sent to Engine ST1.

Tip: Run the `lsengine` command for current Engine status.

## startmetadacheck

Starts the utility that performs a consistency check on the metadata for the entire system or a set of filesets, generates messages in the cluster log, and optionally repairs inconsistencies in the metadata.



## Parameters

### **-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### **-quiet**

Turns off confirmation message for this command.

### **-repair**

Repairs inconsistencies in the metadata and reports the changes in the cluster log. If not specified, repairs are not made.

**Note:** In some cases, manual intervention might be needed to repair the metadata that requires you to take the cluster offline.

### **-scope all | content | structure**

Specifies the scope of the metadata check. You can specify one of the following values:

**all** Checks both the metadata content and structure. This is the default value.

**content** Checks only the metadata content.

**structure** Checks only the metadata structure.

### **-type all | fileset | system**

Specifies the type of metadata to check. You can specify one of the following values:

**all** Checks both the system and fileset metadata. This is the default value.

**fileset** Checks only the fileset metadata.

**system** Checks only the system metadata.

### **-fileset *fileset\_name***

Specifies the names of one or more filesets to check, separated by a comma but no spaces. (Do not include a space after the comma.) If not specified, all filesets are checked.

**Note:** You must specify the **-type fileset** parameter with this parameter to perform a check on a limited set of filesets. You cannot use this parameter with the **-type all** or **-type system** parameters.

## Prerequisites

You must have Administrator privileges to use all parameters in this command. Users with Backup privileges are not allowed to use the **-repair** parameter.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

This command is useful when you suspect inconsistencies in the metadata or after a major restoration of the system.



The metadata checker validates metadata and, if the **-repair** parameter is specified, it also repairs inconsistencies. Check the messages that it generates in the cluster log to determine if there has been a loss of data. You might need to recover lost data from backup or recreate it by reissuing administrative commands.

The metadata checker prints an informational message in the cluster log when it starts and completes, and for each fileset that it processes. It also prints an error or severe message for each problem that it encounters. When the **-repair** parameter is specified, it also prints an error message as it repairs inconsistencies. The command line interface issues either a successful repair or integrity lost message, which indicates that the repair was not successful. Even with a report of a successful repair, you should check the log messages because data recovery or issuing administrative commands might still be necessary.

**Note:** The metadata checker places lost objects in the lost+found directory.

**Note:** When the metadata checker repairs an object in a FlashCopy image, the image is placed in an incomplete state.

**Note:** The metadata checker uses space temporarily in each fileset.

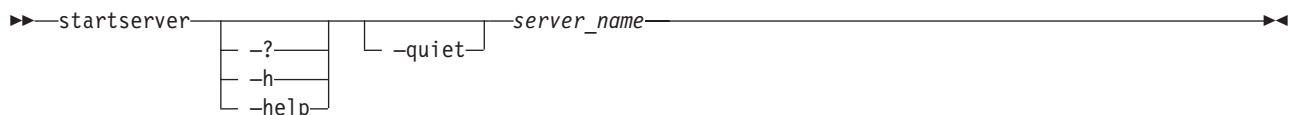
### Example

**Check the system-metadata structure** The following example checks the system-metadata structure and makes necessary repairs:

```
sfscli> startmetadatabasecheck -repair -scope structure -type system
Are you sure you want to start a metadata check process? [y/n] Y
```

## startserver

Starts the specified metadata server.



### Parameters

#### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### -quiet

Turns off confirmation messages for this command.

#### server\_name

Specifies the name a metadata server to start. This metadata server must be part of the cluster.

### Prerequisites

You must have Operator or Administrator privileges to use the command.

## Description

If you run this command from an engine hosting a subordinate metadata server, you can only start the local metadata server. If you run this command from the engine hosting the master metadata server, you can start any metadata server.

If the master metadata server is not running, you cannot start a subordinate metadata server.

When you start the metadata server using the **startserver** command, the subordinate metadata servers in the joining state are placed in their previous state after the master metadata server is up and running. When you start the cluster using the **startcluster** command, all of the subordinate metadata servers are brought to the online state.

The state of the metadata server matches the state of the cluster. Use the **statcluster** command to check on the state of the cluster, or use the **lsserver** command to check on the state of the metadata server.

When you start a metadata server using the **startserver** command, the metadata server restart service on that metadata server is enabled only if metadata server restart service was previously in the standby state.

**Restriction:** If you manually disable the metadata server restart service, and then stop and restart the metadata server, the metadata server restart service will not become enabled.

**Tip:** A message stating that the metadata server started successfully does not necessarily mean that the metadata server is online.

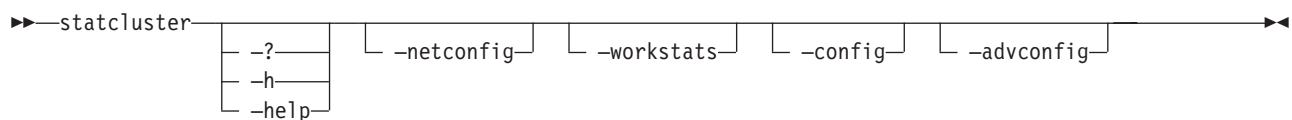
## Example

**Start a metadata servers** The following example starts the metadata server ST1:

```
sfscli> startserver ST1
Are you sure you want to start the server? Starting the server
may cause filesets to be reassigned to this server in accordance with the
fileset assignment algorithm. [y/n] Y
Server ST1 started successfully
```

## statcluster

Displays status, network, workload, and configuration information about the cluster.



## Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

**-netconfig**

Displays network configuration information for the cluster.

**-workstats**

Displays workload statistics related to the total number of transactions, updates, and buffers for the master metadata server.

**-config**

Displays cluster configuration settings.

**-advconfig**

Displays advanced cluster configuration settings.

**Prerequisites**

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

**Description**

If you do not specify a parameter, this command displays the following default information:

- Name of the cluster.
- ID of the cluster.
- Current state of the cluster. See Cluster states for a description for possible values.
- Target state to which the cluster is switching.
- Timestamp of the last current state change.
- Timestamp of the last state change.
- Number of metadata servers in the cluster.
- Number of metadata servers that are actively participating in the cluster (in the offline, online, or quiescent state).
- Committed software version.
- Software version.
- Timestamp when the latest software was committed.
- Software commit status (In Progress or Not In Progress).
- Timestamp of the last installation.

If you specify the **-netconfig** parameter, this command displays the following information in addition to the default information:

- Name of the master metadata server.
- IP address.
- Cluster port number.
- Heartbeat port number.
- Client-metadata server port number.
- Administrative port number.

If you specify the **-workstats** parameter, this command displays the following information in addition to the default information:

- Number of system-update transactions involving the creation, modification, and deletion of system objects.

- Number of system-object transactions, involving the creation, modification, and deletion of objects. The system objects include storage pools, filesets, volumes, policies, and engines.
- Current number of buffers, which are used for system metadata activity, that contain data but are available for reuse (clean buffers).
- Current number of buffers, which are used for system metadata activity, that contain data awaiting I/O to disk (dirty buffers).
- Current number of buffers, which are used for system metadata activity, that are not in use (free buffers).
- Current total number of buffers for system metadata activity.

If you specify the **-config** parameter, this command displays the following information in addition to the default information:

- Pool-space reclamation interval, in minutes.
- List of clients with root or administrative privileges.
- Remote Supervisory Adapter (RSA) user ID.
- RSA password. The password appears as asterisks, with one asterisk per character in the password.
- Size, in 4-KB pages, of the master metadata server buffer.
- Size, in pages, of the subordinate metadata server buffer.
- Administrative process limit.
- Metadata server workload process limit.

If you specify the **-advconfig** parameter, this command displays the following information in addition to the default information:

- Maximum number of missed network heartbeats.
- Network heartbeat interval, in milliseconds.
- Cluster timeout, in milliseconds.
- Maximum number of retries to client.
- Client timeout, in milliseconds.
- Client lease period, in seconds.
- Client-lease-period multiplier.
- Metadata server timeout, in milliseconds.
- Maximum number of missed disk heartbeats.
- Disk heartbeats interval, in milliseconds.
- Available languages.
- Current languages.

### Example

**List network information for the cluster** The following example displays the default and network information for the cluster:

```
sfscli> statcluster -netconfig
Name sanfs
ID 2802
State ONLINE
Target State FULLY QUIESCENT
Last State Change Sep 24, 2003 3:31:52 PM
Last Target State Change Sep 24, 2003 2:40:20 PM
Servers 4
Active Servers 3
```

```

Committed Software Version 1.1.4.2
Software Version 1.00
Last Software Commit Jan 3, 2003 1:40:20 PM
Software Commit Status In Progress
Installation Date Jan 3, 2003 4:56:59 PM

```

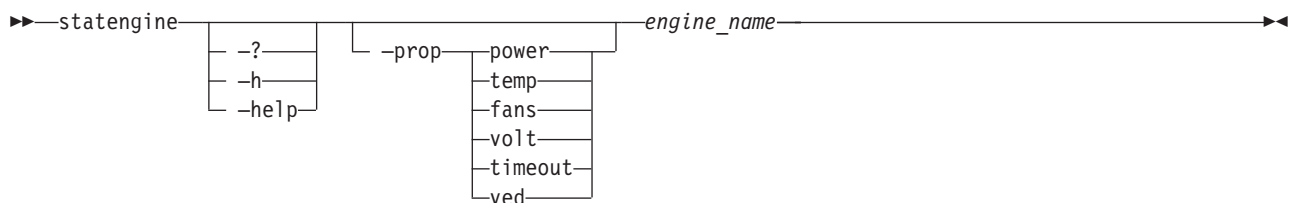
```

=====Network Configuration=====
Master Server ST1
IP 128.0.0.1
Cluster Port 1737
Heartbeat Port 1738
Client-Server Port 1700
Admin Port 1800

```

## statengine

Displays status information about a specific storage engine.



### Parameters

#### **-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### **-prop**

Specifies the property status to display. Possible values are:

**power** Displays power status for the specified engine and operating system.

**temp** Displays temperature status for the components of the specified engine.

**fans** Displays fan status for the specified engine.

**volt** Displays voltage status for the specified engine.

#### **timeouts**

Displays timeout information for the certain hardware and software functions in specified engine environment.

**ved** Displays vital engine data for the specified engine.

#### *engine\_name*

Specifies the name of the storage engine to display.

### Description

If you do not specify a property parameter, this command displays the following default information for each engine:

- Engine IP address.
- Engine name.
- Boot state. Possible values are:

**Before POST**

The engine is powered on but has not started the power-on self test (POST).

**Booted Flash**

The engine has read the System BIOS but has not started loading the operating system.

**Booting OS**

The engine has started but not completed loading the operating system.

**CPUs Held in Reset**

The engine has been reset after a hardware fault.

**In POST**

The engine is running the POST.

**In OS** The engine is running in the normal state.

**Stopped in POST (error detected)**

The engine is powered on but has not completed the POST due to an error.

**Unknown / Power Off**

The boot state is unknown. The engine could be powered off.

- Temperature state. Possible values are:

- The temperature thresholds are not set.

**Normal**

The temperatures of all components in all engines are below the Warning threshold.

**Warning**

The temperature of one or more engine components are above the Warning threshold.

**Error** The temperature of one or more engine components is above than the Soft Shutdown threshold.

**Unknown**

The RSA card could not be accessed.

- Fan state. Possible values are:

**Normal**

All fans are operating above 15% of its fan-speed capacity.

**Warning**

One or more fans are operating below 15% of its fan-speed capacity.

**Error** The RSA card could not be accessed.

- Voltage state. Possible values are:

- The voltage thresholds are not set.

**Normal**

The voltages of all components in all engines are above the Warning Low Voltage threshold and below the Warning High Voltage threshold.

**Warning**

The voltage of one or more engine components is below the Warning Low Voltage threshold or above the Warning High Voltage threshold.

**Error** The RSA card could not be accessed.

If you specify the **-prop power** property parameter, this command displays the following for each engine:

- Engine IP address.
- Engine name.
- Power state (On or Off).
- Number of hours the engine has been powered on.
- Number of times the engine has been restarted. This counter is cleared when the Advanced System Management (ASM) device is cleared to factory defaults.
- Current time on the ASM device's local clock. This time is independent of the time on the engine and is the time used to schedule a power off.

If you specify the **-prop temp** property parameter, this command displays the following for each engine component:

- Engine component. This command displays information for the following components. (Note that the actual component labels and numbers may vary depending on your environment.):

**CPU1**

**CPU2**

**DASD1**

**DASD2**

**Ambient** (overall temperature)

- Temperature state per component.
- Current temperature, in degrees Celsius.
- Warning threshold, in degrees Celsius.
- Soft Shutdown threshold, in degrees Celsius.
- Hard Shutdown threshold, in degrees Celsius.

If you specify the **-prop fans** property parameter, this command displays the following for each engine:

- Fan name. This command displays information for each of the eight fans, labeled Fan 1 through Fan 8.
- Fan state.
- Fan speed as a percentage of the maximum capacity speed.

If you specify the **-prop volt** property parameter, this command displays the following for each engine:

- Engine component. This command displays information for the following components. (Note that the actual component labels and numbers may vary depending on your environment.):

**VRM 1 (1.5 V)**

**VRM 2 (1.5 V)**

**System board (12 V)**

**System board (5 V)**

**System board (3.3 V)**

**System board (2.5 V)**

**System board (1.5 V)**

- Voltage state.
- Voltage.

- Description.
- Warning Low voltage threshold.
- Warning High voltage threshold.

If you specify the **-prop timeouts** property parameter, this command displays the following for each engine:

- POST Watchdog threshold. This is the amount of time the engine will wait for POST to complete before sending an alert and automatically restarting the system. If the threshold is not set, the value is 10 minutes.
- Loader Watchdog threshold. This is amount of time the engine will wait for the operating system to load before sending an alert and automatically restarting the system. If the threshold is not set, the value is 2 hours.
- OS Watchdog timeout. This is the amount of time for the operating system to respond before sending an alert and automatically restarting the system. If the threshold is not set, the value is 4 minutes.
- OS Watchdog check interval. This is the amount of time between the OS Watchdog checks. If the threshold is not set, the value is 4 minutes.
- Power Off Delay threshold. The amount of time to wait for the operating system to shut down before powering off the system. If the threshold is disabled, the value is 30 seconds.

If you specify the **-prop ved** property parameter, this command displays the following for each engine:

- Engine model number.
- Engine serial number.
- Unique universal identifier, which uniquely identifies the engine.
- Firmware revision. This is the revision numbers of the application firmware and startup ROM firmware.
- Logical device's firmware revision date.
- File names of the application and startup ROM firmware.
- Build identifier of the application and startup ROM firmware.

### Example

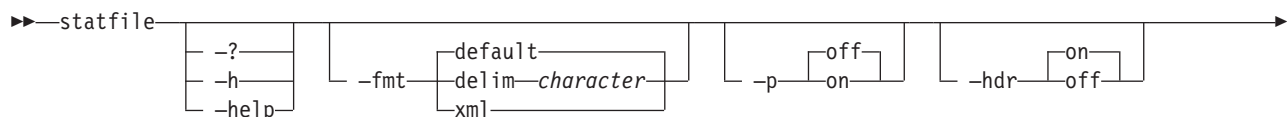
**Display power status for an engine** The following example displays power status information for the engine ST1:

```
sfscli> statengine -prop power ST1
```

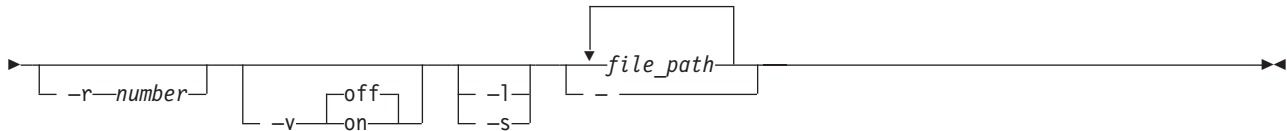
```
Engine IP 39.47.25.19
Name ST1
Power State Off
Power-On Hours 256 hours
Restarts 56
Current ASM Time Aug 12, 2003 4:59:47 PM
```

## statfile

Displays metadata information about the specified files.







## Parameters

### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### -fmt

Specifies the format of the output. You can specify one of the following values:

#### default

Specify this value to display output in a tabular format using spaces as the delimiter between the columns. This is the default value. For example:

```

OS Device Path LUN ID Engine WNN Vendor Product
=====
/dev/sda 2 12:34:56:78:91:23:45:67 IBM 2105F20

Size (MB) Volume State
=====
 47000 Vol1 Assigned

```

#### delim *character*

Specify this value to display output in a tabular format using the specified character to separate the columns. If you use a shell metacharacter (for example, \* or \t) as the delimiting character, enclose the character in single quotation marks (') or double quotation marks ("). A blank space is not a valid character. For example:

```
/dev/sda,2,12:34:56:78:91:23:45:67,IBM,2105F20,47000,Vol1,Assigned
```

#### xml

Specify this value to display output using XML format, for example:

```

<IRETURNVALUE>
<INSTANCE CLASSNAME="STC_AvailableLUNs">
<PROPERTY NAME="LUNID" TYPE="uint64"><VALUE TYPE="uint64">2</VALUE>
</PROPERTY>
<PROPERTY NAME="NodeWWN" TYPE="string"><VALUE TYPE="string">
12&58;34&58;56&58;78&58;91&58;23&58;45&58;67</VALUE></PROPERTY>
<PROPERTY NAME="Vendor" TYPE="string"><VALUE TYPE="string">IBM</VALUE>
</PROPERTY>
<PROPERTY NAME="Product" TYPE="string"><VALUE TYPE="string">2105F20
</VALUE></PROPERTY>
<PROPERTY NAME="Size" TYPE="uint64"><VALUE TYPE="uint64">47000</VALUE>
</PROPERTY>
<PROPERTY NAME="State" TYPE="uint64"><VALUE TYPE="uint64">2</VALUE>
</PROPERTY>
<PROPERTY NAME="VolumeName" TYPE="string"><VALUE TYPE="string"></VALUE>
</PROPERTY>
<PROPERTY NAME="DeviceID" TYPE="string"><VALUE TYPE="string">
&47;sda</VALUE></PROPERTY>
</INSTANCE>
</IRETURNVALUE>

```

### -p

Specifies whether to display one page of text at a time or all text at once.

**off** Displays all text at one time. This is the default value when the **sfscli** command is run in single-shot mode.

**on** Displays one page of text at time. Pressing any key displays the next page. This is the default value when the **sfscli** command is run in interactive mode.

**-hdr**

Specifies whether to display the table header.

**on** Displays the table header. This is the default value.

**off** Does not display the table header.

**-r** *number*

Specifies the number of rows per page to display when the **-p** parameter is on. The default is 24 rows. You can specify a value from 1 to 100.

**-v** Specifies whether to enable verbose mode.

**off** Disables verbose mode. This is the default value.

**on** Enables verbose mode.

**-l** Displays default information about the logical units (LUs).

**-s** Displays default information about the LUs.

*file\_path*

Specifies one or more file paths referenced from the global fileset for which to display information.

**Restriction:** Wildcards are not allowed in the file path.

- Specifies that you want to read the file paths from stdin (for example, **- << /work/path\_list.txt**).

## Prerequisites

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

No data is available if the fileset is detached.

You can use this command to ensure that a specific file was correctly placed into a storage pool when it was created. This helps to validate policy rules. You can also use this command to validate metadata information viewed through a client.

After a metadata server failure and recover, there is a grace period where new locks for metadata are not granted. If the fileset is in a grace period, the fileset cannot access the metadata or file information, and this command fails.

This command displays information about files, not directories or symbolic links.

If you do not specify a listing parameter, this command displays the following default information for each file:

- File name.
- Storage pool.
- Fileset.
- Metadata server.
- File size, in megabytes.
- Date and time that the file was last modified.

If you specify the **-l** listing parameter, this command displays the following information for each file in addition to the default information:

- File type (Regular File, Directory, Symbolic Link, FIFO or unknown).
- File block size, in bytes.
- Number of blocks allocated to the file.
- Number of names that reference this file.
- Date and time that the file was created.
- Date and time that the metadata was last modified without altering the file data (for example, moving, touching or changing permissions will affect the metadata but not the file data).
- Date and time that the file was last accessed.

If you specify the **-s** listing parameter, this command displays the following information for each file:

- File name.
- Storage pool.
- Fileset.

### Example

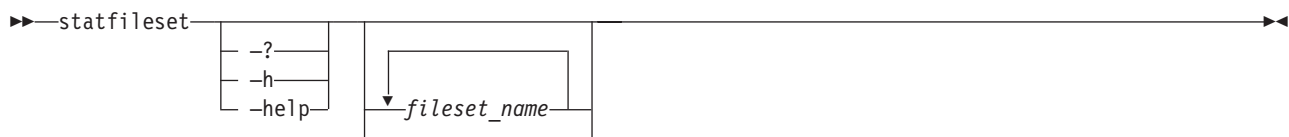
**List file information** The following example displays information about files located in path `sanfs/fileset_AP/notes/tips.nsf`:

```
sfscli> statfile sanfs/fileset_AP/notes/tips.nsf
Name Pool Fileset Size (MB) Server
=====
resources.txt DEFAULT_POOL FS1 0.7 ST1
sales.lwp pool_1 OS 3.6 ST2

File Modified
=====
Oct 15, 2003 1:22:22PM
Oct 25, 2003 5:19:03PM
```

## statfileset

Displays the transaction statistics for one or more specified filesets.



### Parameters

**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

*filesset\_name*

Specifies the names of one or more filesets whose statistics you want to view.

- Specifies that you want to read the names of one or more filesets from stdin (for example, `- << /work/cnt_list.txt`).

Description

If you run this command from an engine hosting a subordinate metadata server, you can only view fileset statistics associated with the local metadata server. If you run this command from the engine hosting the master metadata server, you can view fileset statistics associated with all metadata servers in the cluster.

A *transaction* is a work request that is handled by the metadata server. You can use the number of transactions performed by each fileset on the metadata server to aid in balancing the workload among all of the metadata servers in the cluster.

This command displays the following information for each fileset:

- Name of the fileset.
- Name of the metadata server serving the fileset.
- Number of transactions that are in progress.
- Number of transactions that have been stopped.
- Number of transactions that have been retried.
- Number of transactions that have been initiated since the metadata server started (including those transactions that are initiated during startup). This count includes the transactions that are in progress, stopped, and completed.
- Number of transactions that completed successfully.

Example

**Displaying number of transactions** The following example displays the number of started and completed transactions for filesets being served by the local metadata server:

sfscli> statfileset

Name	Server	Current Transactions	Stopped	Retried	Started	Completed
cnt_A	ST1	12344	234	344	908798	87656
cnt_B	ST1	234445	21344	3444	1098768	987689

statldap

Displays Lightweight Directory Access Protocol (LDAP) configuration settings.



Parameters

-? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

Prerequisites

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

This command displays the following information:

- IP address of the LDAP server.
- Administrative LDAP ID that is used to make authentication queries. This is the user ID that is used to sign on to the LDAP server, not the user IDs that LDAP creates for SAN File System.
- Password corresponding to the Administrative LDAP ID. The password appears as asterisks, with one asterisk per character in the password.
- Secure connection (on or off). If on, the connection is made using Secure Sockets Layer (SSL).
- Base distinguished names (DN) where roles are located in the LDAP server.
- LDAP attribute containing the login name.
- LDAP attribute containing the role's name.
- LDAP attribute containing the role's members.
- Maximum age, in seconds, of items in LDAP cache

## Example

**Display LDAP settings** The following example lists the LDAP configuration settings:

```
#sfsccli statldap
LDAP Server IP 123.123.123
Administrative LDAP User ldapadmin
Administrative LDAP Password *****
Secure Connection Off
Roles DN ou=Roles,o=0z
User ID Attribute uid
Role ID Attribute cn
Role Member Attribute role0ccupant
Cache Timeout 600 secs
```

## statpolicy

Displays policy statistics for each fileset.



## Parameters

### -? | -h | -help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### -pool

Displays policy statistics for all filesets for all storage pools.

### -rule

Displays policy statistics for all filesets for all rules.

## Prerequisites

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

Policy statistics are maintained for each fileset. These statistics reset when:

- The cluster is stopped and restarted.
- A new policy is activated.
- A metadata server is stopped, started, restarted, added, or dropped.
- A fileset is moved to another metadata server or detached.

You can manually reset the counters for the statistics by reactivating the current policy.

No data is available about a fileset, if the fileset is detached.

If you specify the **-pool** parameter, this command displays the following information:

- Pool name.
- Number of times a file was placed into this storage pool
- Last time a file was placed into this storage pool.

If you specify the **-rule** parameter, this command displays the following information:

- Rule name.
- Ordinal position of the rule. The order is not viewable when editing a policy. It is sequenced automatically by the system. This is important because rules are evaluated in order. If possible, initially sort the table from lowest number to highest number
- Number of times that the rule has caused an error while being evaluated. These are not syntax errors.
- Number of times that the rule has been evaluated and its conditions were false, so the rule was not applied.
- Number of times that the rule has been evaluated and its conditions were true, so that the rule was applied.
- Last time a rule was applied. This is the last time the rule placed a file into a storage pool.

## Example

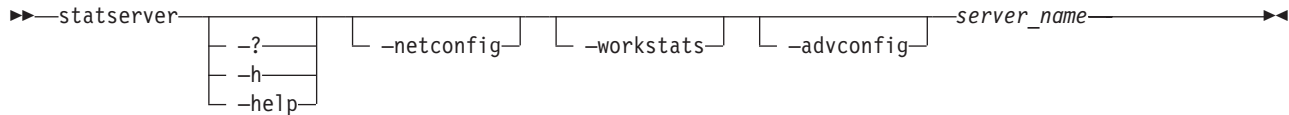
**List policy statistics** The following example displays the policy rule statistics:

```
sfscli> statpolicy -rule
Rule Name Position Evaluation Evaluations Applied
 Errors Not Applied Evaluations
=====
Myrule 1 0 97 87
Myotherrule 2 23 798 8098

 Last Applied
=====
Oct 28, 2003 12:35:43PM
Oct 28, 2003 12:35:43PM
```

## statserver

Displays status, configuration, and workload information for a specific metadata server in the cluster, if the command is issued from the master metadata server. Lists status, configuration, and workload information for the local metadata server, if the command is issued from a subordinate metadata server.



### Parameters

#### `-? | -h | -help`

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### `-netconfig`

Displays network configuration information for the specified metadata server.

#### `-workstats`

Displays statistical information for the local metadata server.

#### `-advconfig`

Displays advanced settings, which require a reboot when changed, for the specified metadata server.

#### *server\_name*

Specifies the metadata server to display. If you issue this command from a subordinate metadata server, this must be the name of the local metadata server.

### Description

If you run this command from an engine hosting a subordinate metadata server, you can display information about only the local metadata server. If you run this command from the engine hosting the master metadata server, you can display information about any metadata server.

The *server\_name* parameter must be specified. If you do not specify the `-netconfig`, `-workstats`, or `-advconfig` parameter, the following statistics are displayed:

- Metadata server name.
- Role of the metadata server (Master or Subordinate).
- Pending software version.

If you specify the `-netconfig` parameter, the following statistics are displayed in addition to the default statistics:

- Master metadata server name.
- Metadata server-to-metadata server transport protocol (UDP)
- Client-to-metadata server transport protocol (TCP)
- IP address.
- Cluster port number.
- Heartbeat port number.
- Client-server port number.
- Administrative port number.

If you specify the **-workstats** parameter, the following statistics are displayed in addition to the default statistics:

- Number of file-system-update transactions, that involve creating, modifying, and deleting system objects.
- Number of file-system transactions, that involve reading, creating, modifying, and deleting objects. The system objects include storage pools, filesets, volumes, policies, and engines.
- Number of buffers that contain data awaiting I/O to disk (dirty buffers).
- Number of buffers that contain data but are available for reuse (clean buffers).
- Number of buffers that are not in use (free buffers).
- Total number of available buffers.
- Number of session locks held by the engine hosting the metadata server. Clients acquire session locks to perform file management operations.
- Number of data locks held by the engine hosting the metadata server. Clients hold data locks to cache data pages and attributes of files and to cache read-only attributes and contents of directories and links.
- Number of byte-range locks held by the engine hosting the metadata server. Clients use byte-range locks to implement POSIX, SYSV and Berkeley lock system calls.

If you specify the **-advconfig** parameter, the following statistics are displayed in addition to the default statistics:

- File-space reclamation process limit.
- Synchronous commit mode:
  - sync** Schedules and buffers metadata changes instead of performing an immediate write.
  - async** Metadata changes are written to disk immediately.
- List of current profile parameters being used.
- List of current trace parameters being used as output.
- Time interval, in seconds, at which the user and system-time statistics are gathered. A value of 0 indicates that this property is disabled.

## Example

**List information about the metadata server** The following example displays all information about the server:

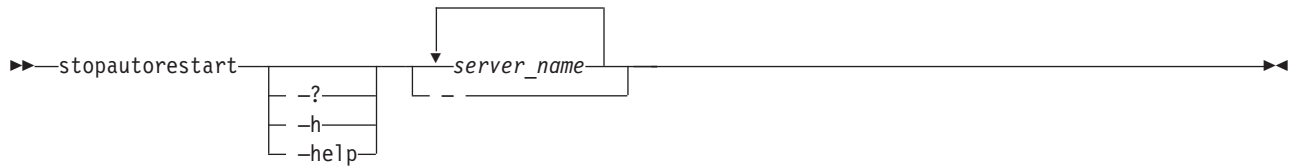
```
sfsccli> statserver -netconfig ST1
```

Server	ST1
Server Role	Subordinate
Most Current Software Version	V1.003
=====Network Configuration=====	
Master Server	ST4
Server-to-Server Transport Protocol	UDP
Client-to-Server Transport Protocol	TCP
IP	128.0.0.1
Cluster Port	1737
Heartbeat Port	1738
Client-Server Port	1700
Admin Port	1800



## stopautorestart

Disables the metadata server from restart automatically if it is down.



### Parameters

#### `-? | -h | -help`

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### `server_name`

Specifies the names of one or more metadata servers to disable from restarting automatically.

- Specifies that you want to read the names of one or more metadata servers to enable from restarting automatically from stdin (for example, `- << /work/server_list.txt`).

### Prerequisites

You must have Administrator privileges to use the command.

### Description

**Note:** If you run this command from an engine hosting a subordinate metadata server, you can stop the metadata server restart service on only the local metadata server. If you run this command from the engine hosting the master metadata server, you can stop the metadata server restart service on any metadata server.

### Example

**Disable the automatic-restart service** The following example disables the automatic-restart service for metadata server ST1:

```
sfsccli> stopautorestart ST1
```

The automatic restart service for server ST1 successfully disabled.

## stopcluster

Stops all metadata servers in the cluster gracefully.



## Parameters

### **-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### **-quiet**

Turns off confirmation messages for this command.

## Prerequisites

You must have Operator or Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

This command does not shut down the engine on which the specified metadata server runs.

When you stop the master metadata server using the **stopserver** command, the subordinate metadata servers are abruptly moved to the joining state. When you stop all of the metadata servers in the cluster using the **stopcluster** command, all of metadata servers are brought down gracefully.

**Tip:** Before you use the **stopcluster** command, you can quiesce the cluster. Use the **quiescecluster** command to change the state of all metadata servers in the cluster to one of three quiescent states.

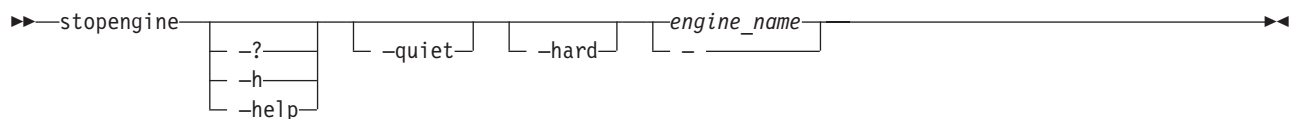
## Example

**Stop the cluster** The following example stops the cluster gracefully:

```
sfscli> stopcluster
Are you sure you want to shut down the cluster? [y/n] Y
Cluster shut down gracefully.
```

## stopengine

Shuts down the operating system and powers off one or more storage engines.



## Parameters

### **-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### **-hard**

Powers off the specified engines without first shutting down the operating system. If not specified, this command shuts down the operating system before powering off the storage engines.

Turns off confirmation messages for this command.

Specifies the name of the storage engines to power off.

- ### Prerequisites

### Description

An engine is not powered off until the time that is indicated by the `PowerOff` delay threshold property (displayed by the **statengine** command). Wait this amount of time before you run another **stopengine** or **startengine** command.

**Power off an engine** The following example shuts down the operating system and then powers off engine ST1:

```
sfscli> stopengine ST1
Are you sure you want to stop engine ST1? [y/n] Y
A start request has been sent to Engine ST1.
Tip: Run the lsengine command for current Engine status.
```

Stops the metadata check utility that is currently in progress.



**-? | -h | -help**

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

Turns off confirmation messages for this command.

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Example

**Stop the metadata check process** The following example stops the current metadata check process:

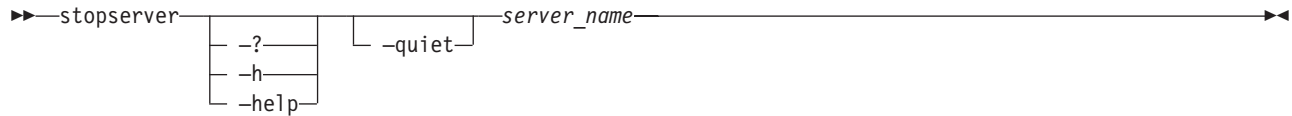
```
sfsccli> stopmetadatacheck
```

Are you sure you want to stop the metadata check process? y/n Y

The metadata check is stopping. This may take a few minutes to complete.

## stopserver

Shuts down a subordinate metadata server gracefully.



### Parameters

#### `-? | -h | -help`

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### `-quiet`

Turns off confirmation messages for this command.

#### `server_name`

Specifies the name of the subordinate metadata server to shut down. This metadata server must be part of the cluster.

### Prerequisites

You must have Operator or Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

### Description

To use this command, you must be logged into the master metadata server and the master metadata server must be running.

This command does not shut down the engine on which the specified metadata server runs and does not remove the metadata server from the cluster.

When you stop the master metadata server using the **stopserver** command, the subordinate metadata servers are abruptly moved to the JOINING state. When you stop all of the metadata server in the cluster using the **stopcluster** command, all of metadata servers are brought down gracefully.

When you stop a metadata server using the **stopserver** command, the metadata server restart service on that metadata server is changed to the STANDBY state only if metadata server restart service was previously in the ON state.

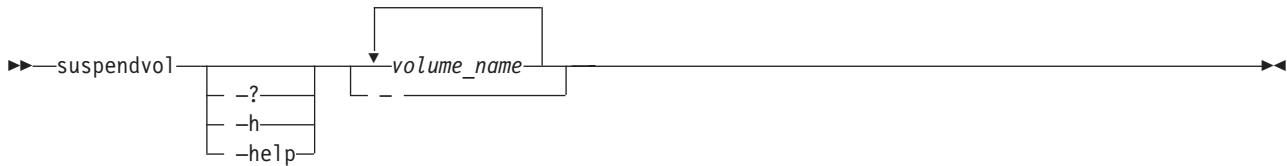
## Example

**Stop a metadata server** The following example stops the metadata server ST1:

```
sfsccli> stopserver ST1
Are you sure you want to stop the server and distribute its workload to
the remaining servers? [y/n] Y
Server ST1 shutdown successfully.
```

## suspendvol

Suspends one or more volumes so that the metadata server cannot allocate new data on the volumes.



### Parameters

`-? | -h | -help`

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

*volume\_name*

Specifies the names of one or more volumes to suspend.

- Specifies that you want to read the names of one or more volumes to suspend from stdin (for example, `- << /work/vol_list.txt`).

### Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

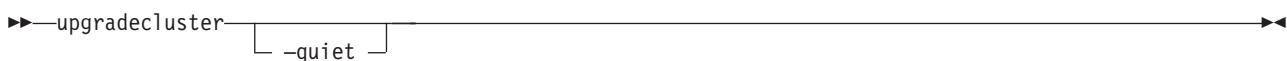
## Example

**Suspend a volume** The following example suspends activity on volume vol2:

```
sfsccli> suspendvol vol2
Volume vol2 suspended.
```

## upgradeclass

Verifies that all the engines in the cluster have been upgraded to the new version of the software and then initiates the cluster upgrade process.



### Parameters

`-quiet`

Turns off confirmation messages for this command.

## Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Description

This command only works after all engines have been upgraded to the same software version. All clients identified to the master metadata server should be running version 2.1 software.

Any version 1.1 clients still identified with a subordinate metadata server will have their leases expired and any attempt to reidentify the client will result in an error.

You do not need to stop the cluster to perform the upgrade.

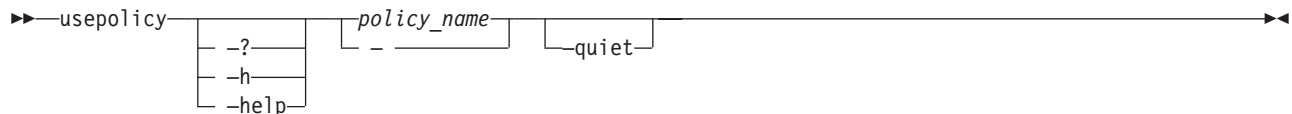
## Example

**Upgrade the cluster** The following example upgrades the cluster:

```
sfscli> upgradecluster
Are you sure you want to upgrade the cluster software? [y/n] Y
Cluster upgrade successful.
```

## usepolicy

Directs the metadata server to make an existing policy the active policy and applies its rules to all subsequent file creations.



## Parameters

### `-? | -h | -help`

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

### `policy_name`

Specifies the name of the policy to activate.

- Specifies that you want to read the name of the policy to activate from stdin (for example, `– << /work/policies_list.txt`).

### `-quiet`

Turns off confirmation messages for this command.

## Prerequisites

You must have Administrator privileges to use the command.

You must be logged in to the operating system on the engine hosting the master metadata server to run this command.

## Example

**Activate a policy** The following example activates the policy *test\_policy*:

```
sfscli> usepolicy test_policy
Are you sure want to use this policy? New files
should be allocated to a pool that is accessible to the
clients where the file is needed. [Y/N] Y
Policy test_policy is now the active policy.
```

## Client commands

There is a set of commands for each client operating system that SAN File System supports.

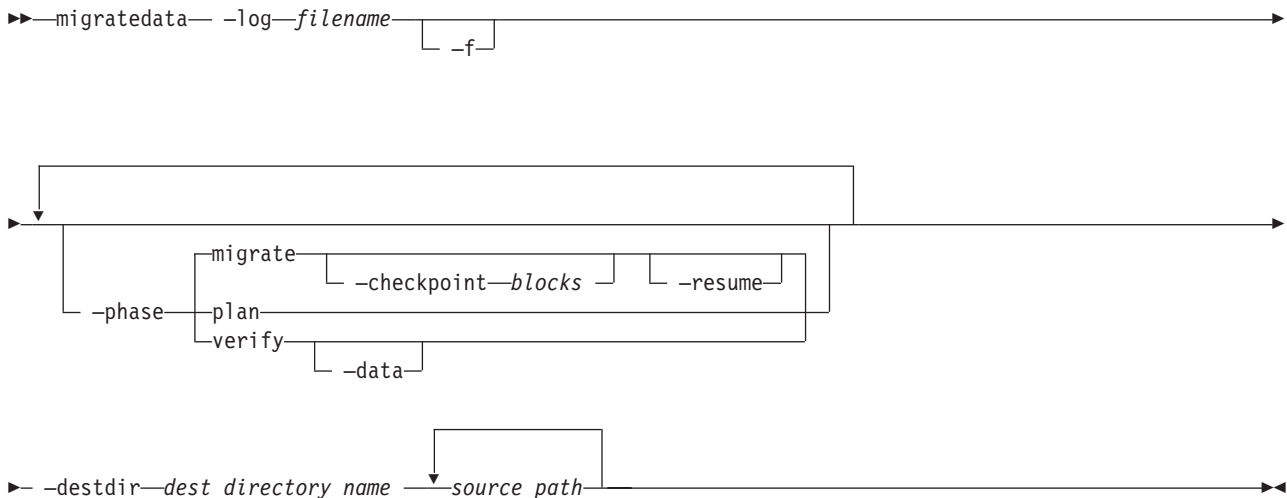
### Common client commands

This topic provides a brief description for each command that has the same syntax on multiple operating systems.

Command	Operating System	Description
<b>Migration</b>		
"migratedata"	AIX, Linux, Solaris, and Windows	Migrates data to SAN File System.
<b>Virtual client setup and removal</b>		
"rmstclient" on page 267	AIX, Linux, and Solaris	Unmounts the global namespace, removes the SAN File System client, and unloads the file-system driver.
"setupstclient" on page 267	AIX, Linux, and Solaris	Configures and starts a client.

### migratedata

Migrates data to SAN File System.



### Parameters

**-log filename**

Specifies the location of a file in which to log migration activities, warnings,

and errors. When used with the **-plan migrate -resume** parameter, the **-log** parameter specifies the log file from which to read information about the last completed block or file.

**Attention:** You must specify the correct log file with migrate **-resume** and verify that the source and destination directories specified on the command line match those in the log file.

- If you specify an incorrect log file and the **-f** parameter, **-resume** displays a warning and overwrites the target file-system data with wrong information.
  - If you specify an incorrect log file, but do not specify the **-f** parameter, this command displays an error and exits.
- f** Specifies that the migration should continue even if there is an error with a file. If specified with the **-phase migrate** parameter, this command skips any files with errors, but continues with the migration process. If not specified, an error results in the entire migration being stopped before the file that caused the error. You can then restart the migration after fixing the error.

If specified with the **-phase verify** parameter, this command adjusts any missing metadata attributes, such as permissions and times. If there is a mismatch in size, however, this command will not try to readjust the metadata attributes.

#### **-phase**

Specifies the migration phase to run. Choices include:

**plan** Gathers information about the available system resources (available memory, number of CPUs, size of the source tree and space available on the destination file system), copies sample files from source directory to estimate transfer rates, and provides an estimated time for the migration of the data set.

#### **migrate**

Reads data from source file system and writes the data to the destination file system. Although not required, for large data sets, you should run this command in planning mode first. You can stop the migration process at any point and resume from the last completed file or block (using the **-resume** parameter).

This is the default value.

**verify** Verifies the integrity of the migrated data using the Message Digest 5 verification algorithm on the contents of the file, as well as verifying consistency of the metadata (such as owner and modification time stamp settings) between the source and destination files.

You can specify more than one phase. For example, to plan, migrate, and verify the data, specify **-phase plan -phase migrate -phase verify**. Although you can specify the phases in any order, this command always estimates the completion time, migrates data, and then verifies the migrated data.

If the **-phase** parameter is not specified, this command runs only the migration phase.

#### **-checkpoint blocks**

Shows the progress when migrating large files. If you specify this parameter, the **migratedata** command writes a checkpoint in the log file after each specified number of blocks of a file has been migrated. (The block size depends on the client platform.) For example, if you specify **-checkpoint 20**, this command makes an entry in the log file each time 20 blocks of file data is



migrated. On a platform with a block size of 16 MB, this command writes to the log file after each 320 MB of data from a file has been migrated.

If the migration process is interrupted, this parameter allows you to resume the migration at the place it left off.

If unspecified, the **migratedata** command makes an entry in the log file after each complete file has been migrated. You can resume the migration at the point of the last migrated file.

**-resume**

Resumes the migration from the last completed block or file (logged in the log file specified by the **-log** parameter). If the log file indicates that some files in the source directory are migrated and this parameter is not specified, this command restarts the migration process from the beginning (performs a fresh migration).

**-data**

Verifies every block of source data (file data and metadata) with the destination data. If not specified, this command verifies only the metadata unless there is a mismatch in the file attributes, in which case this command then verifies the file data.

**Note:** Verifying all data is very time consuming and can take as long as the migration itself.

**-destdir** *dest\_directory\_name*

Specifies the name of the destination directory for the migrated data. The directory can either exist or be a new directory name. IBM recommends that you create the directory before beginning the migration process. If the directory does not exist, this command creates the directory using the default permissions.

*source\_path*

Specifies one or more paths of directories or files to migrate.

**Prerequisites**

You must have root privileges on a UNIX-based client or Administrative privileges on Windows to use this command.

You must run this command from the `/usr/tank/migration/bin` directory.

All storage pools, all filesets, and at least one policy must be set up. All activity (from applications, such as database servers and application servers, or users) that modifies data on the source and destination file systems must be stopped and remain stopped to guarantee consistency of the migrated data.

The destination directory must exist with correct set of permissions and appropriate storage policies must be configured.

**Example**

**Migrating data** This example migrates data from the `work/capital` directory on the client machine to the `sanfs/cnt1` directory in the global namespace. A checkpoint is written to the `mgrt_capital.log` log file each time 20 blocks of file data is migrated.

```
migratedata -log /mgrtlogs/mgrt_capital.log -phase migrate -checkpoint 20
-destdir /mnt/tank/sanfs/cnt1 work/capital
```

## rmstclient

Unmounts the global namespace, removes the virtual client, and unloads the file system driver from the local client machine.



### Parameters

#### **-prompt**

Prompts for required parameters, using values from the configuration file, if available.

#### **-noprompt**

Runs silently, using parameters from the configuration file (/usr/tank/client/config/stclient.conf). If a required parameter is not available, the command exists with an error.

### Prerequisites

You must have root privileges to use this command.

You must unmount the SAN File System before invoking this command.

### Example

**Remove a client** The following example removes the local SAN File System client without prompting:

```
rmstclient -noprompt
```

## setupstclient

Configures and starts SAN File System clients.



### Parameters

#### **-prompt**

Forces the **setupstclient** command to prompt for all configuration values.

#### **-noprompt**

Runs silently, using parameters from the configuration file. If the configuration file does not exist or if a required parameter is not available or invalid, the command exits with an error.

### Prerequisites

You must have root privileges to use this command.

### Description

This command configures and starts, or restarts a SAN File System client.

If you do not specify a parameter, this command runs silently using values from the configuration file as defaults. It only prompts for any required information, if a configuration file does not exist or if a value in an existing configuration file is not valid.

This command maintains any values given by the user in a configuration file in parameter=value format. The default configuration file is /usr/tank/client/config/stclient.conf.

Specify the -prompt parameter to force the command to prompt for all configuration values. In this case, if a configuration file exists, the command presents the value from the configuration file as the suggested default when the command displays a prompt. If a configuration file does not exist, the command presents the manufacturing default as the suggested default.

If you specify the -noprompt parameter, the command expects the configuration file to exist. If the file does not contain valid values, the command exits with an error.

### Example

**Setup a client** The following example configures and starts SAN File System clients:

```
setupstclient
```

## AIX-client commands

This topic provides a brief description for each AIX-client command.

**Note:** You must have root privileges to use these commands.

Command	Description
<b>Migration</b>	
"migratedata" on page 264	Migrates data to SAN File System.
<b>Status</b>	
"stfsstatus" on page 276	Displays the version of the file-system driver for the specified virtual client.
<b>Volumes and LUNs</b>	
"stfsdisk" on page 271	Scans the SAN File System for new and removed volumes.
<b>Virtual client setup and removal</b>	
"rmstclient" on page 267	Unmounts the global namespace, removes the SAN File System client, and unloads the file-system driver.
"setupstclient" on page 267	Configures and starts a client.
"stfsclient" on page 269	Creates or destroys a virtual client.
"stfsdriver" on page 273	Loads the file-system driver as a kernel extension.
"stfsmount" on page 275	Mounts the global namespace.

Command	Description
"stfsmount" on page 277	Unmounts the global namespace.

## stfsclient

Creates or destroys a virtual client.

```

▶▶ stfsclient -create [client_name] [server_name server_IP_address] [:port]
▶ -kmname kernel_ext_name -converter 8859-1 [-quiet]

```

or

```

▶▶ stfsclient -destroy [client_name] -kmname kernel_ext_name
▶ [-quiet]

```

### Parameters

#### -create

Creates a new virtual client.

#### -destroy

Destroys an existing virtual client.

#### *client\_name*

Identifies the unique name of the virtual client that you want to create or destroy. The default client name is the host name of the client system.

#### *server\_name*

Specifies the host name of a metadata server in the SAN File System. The metadata server that you specify informs the global namespace image of all other metadata servers.

This parameter is not required if this is not the first mount for a particular virtual client.

#### *server\_IP\_address*

Specifies the IP address, in dotted decimal notation, of a metadata server in the SAN File System.

#### *port*

Specifies the port number of the specified metadata server. The default is 1700.

#### -kmname *kernel\_ext\_name*

Identifies kernel-extension name of the file-system-driver instance associated with the virtual client.

The file-system driver is loaded as a kernel extension. To identify the instance of the file-system driver, you identify the kernel extension. The kernel-extension name is the same as name and location of the file-system driver that was used to load the driver (for example, /usr/tank/client/bin/stfs for AIX).

**-devices**

Determines which devices (also called disks or LUNs) that the virtual client considers as SAN File System volumes. The default is the value of the STFS\_DEVICES environment variable or, if that is not set, `"-devices=pat=/dev/rhdisk*`.

In addition to creating the virtual client, this command discovers which disks, or candidates, are available to the virtual client as volumes and transmits the candidate list to the virtual client. The **-devices** parameter controls the candidates list.

**dir=directory**

The candidates list is made up of those devices that have device special files in the specified directory (for example: `-devices=dir=/dev/stfsdisk`).

The easiest way to mount the global namespace is to specify `-devices=pat=/dev/rhdisk*`, which looks at every SCSI-disk-like device in the system and whatever looks like a SAN File System disk is accessed when the metadata server refers to that disk's SAN File System disk identifier.

If you want the client to be more selective about what disks it considers available, you can create a `/dev/stfsdisk` directory, put device-special files (or symbolic links) for your candidates in it, and just let `-devices=dir=/dev/stfsdisk` default.

**pat=pattern**

The candidates list is made up of those devices that have device-special files whose file specifications match the specified pattern. You can use \* wildcards in the last (filename) component but not in the directory components (for example, `-devices=pat=/dev/rhdisk*`).

**none** The candidates list is empty. Use this value when you want to establish the candidate list with a separate command, perhaps using a selection method more sophisticated than the `stfsclient` command offers.

**-quiet**

Turns off informational messages for this command. This parameter does not affect error messages.

**Prerequisites**

You must have root privileges to use this command.

**Description**

This command creates or destroys a virtual client. A *virtual client* is an entity that communicates with a metadata server and, indirectly, with other SAN File System client. In this release, only one virtual client is supported per client machine. The terms virtual client and client can be used interchangeably.

A virtual client is associated with exactly one SAN File System. There is one file cache and one set of disk candidates per virtual client. Each virtual client running on the same system is as separate as if it were running on a different system. They share nothing except the file-system drive code that they execute.

A SAN File System virtual client is uniquely identified in the context of its file-system driver, and in the context of its SAN File System, by its client name.

To use the files in a global namespace, the virtual client must have a global namespace image. Creating a global namespace image makes the directory structure in the global namespace appear in the client’s file structure. To create a global namespace image, use the **stfsmount** command.

A client can access and create data that is stored in a global namespace. Each virtual client can access data on multiple images in the same global namespace.

The client considers a file to be one file even if it appears with two different file names in two different global namespace images.

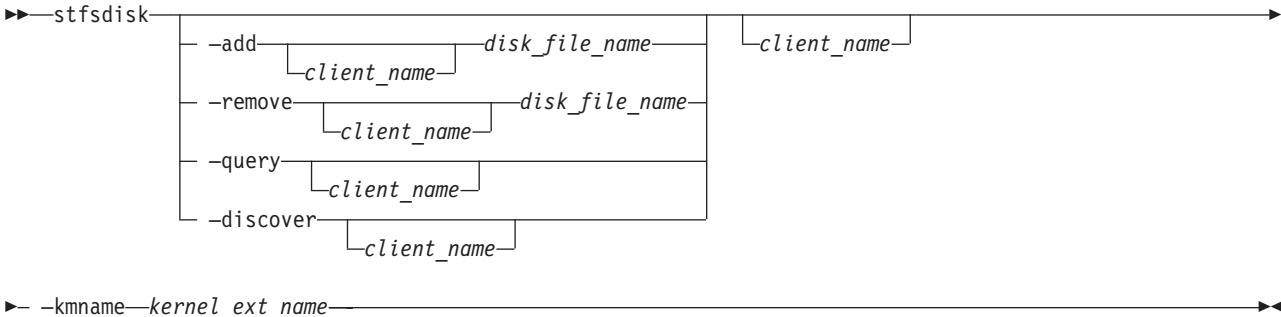
**Example**

**Create a virtual client** The following example creates a virtual client:

```
stfsclient -create MDS1:1700 -kmname /usr/tank/client/bin/stfs
-converter 8859-1
```

**stfstdisk**

Controls the SAN File System volumes (disks) that a client can access.



**Parameters**

**-add**

Adds the specified disk-specific file name to the disk-candidate list. If the disk-specific file name is already in the list, the client performs a disk discovery procedure on it, updating its database if indicated.

**-remove**

Removes the specified disk-specific file name from the disk candidate list. If the disk-specific file name is not in the list, this command does nothing, but does not consider it an error.

**-query**

Displays the list of disk-specific file names in the current disk-candidate list and the status of each. Possible status values are:

**ACTIVE**

Indicates that the disk is a valid SAN File System user-data volume and is available to the client to perform file reads and write operations.

**INACTIVE**

Indicates that the disk is not a valid SAN File System user-data volume and is not available to the client to perform file reads and writes. A

disk can be inactive if there were I/O errors when the client tried to access the disk, if the disk does not contain a SAN File System label, or if the disk's SAN File System label says it is something other than a user-data volume.

**-discover**

Rebuild the database of usable disks by going through the current candidate-disk list and attempting to access each disk, determine if it is a valid SAN File System user-data volume, and read its SAN File System global disk ID. If a disk has become accessible or inaccessible, or changed its identity since the last time this disk-discovery procedure was run, the virtual client updates its candidate-disk list accordingly.

This parameter causes the disk-discovery procedure to start. The procedure typically ends before the disk-discovery procedure completes. While the disk-discovery procedures are in progress, any file-system access that would fail because the virtual client cannot find a specific disk will wait until the disk-discovery procedure completes, and then proceed on the basis of the new disk-accessibility information.

*disk\_file\_name*

Specifies the file name of the disk to add to or remove from the disk candidate list. This must be a raw disk file such as /dev/vpath0, not a logical volume file such as /dev/hdisk0.

*client\_name*

Specifies the name of the virtual client whose disk-access you are controlling.

**-kmname** *kernel\_ext\_name*

Identifies kernel-extension name of the file-system-driver instance associated with the client.

The file-system driver is loaded as a kernel extension. To identify the instance of the file-system drive, you identify the kernel extension. The kernel-extension name is the same as the name and location of the file-system driver that was used to load the driver (for example, /usr/tank/client/bin/stfs for AIX). Note that the kernel extension name might not be unique.

**Prerequisites**

You must have root privileges to use this command.

**Description**

A client reads and writes files by accessing the disks on which the file data resides. To control which disks that a client can access, SAN File System identifies that disk by a SAN File System global disk identifier, and the disk-access subsystem associates that identifier with the name that the AIX operating system uses to identify that disk. The disk-access subsystem maintains a database that correlates global-disk identifiers with AIX device numbers. When the client needs to access a data block of a file, it consults that database.

The disk-access subsystem maintains the database by reading certain disks at certain times and looking for a SAN File System global disk identifier. If it finds the identifier, it determines whether the disk is a SAN File System user-data volume. If the disk is a volume, it adds the disk to its database.

The set of disks that the disk-access subsystem searches is called the *disk-candidate list*. The **stfsclient** command creates the disk-candidate list when it creates the virtual client. You can modify the list using the **-add** and **-remove** parameters.

The candidate-disk list is a list of unique disk-special file names. Because a disk can be referred to by more than one disk-special file name, the list is not strictly a list of unique devices. Actually examining disks and updating the database of valid user-data volumes is separate from maintaining the candidate-disk list.

When you add a disk to the candidate-disk list, the client immediately tries to read it and adds it to the database. But the disk becomes and stays a candidate regardless of the results of that operation.

You can force the client to rescan the entire list of candidate disks using the **-discover** parameter. The client updates its database of user-data volumes according to the results of this discovery, adding and removing disks as necessary. The results of the discovery do not affect the candidate-disk list, however.

Note that device file names can change as the client runs. Such a change has no effect on the client unless something causes a disk-discovery procedure to run. For example, if you add /dev/rhdisk35 as a candidate disk, and the client successfully identifies it as a SAN File System user-data volume, and then you delete /dev/rhdisk35, the client continues accessing that disk as before. The disk /dev/rhdisk35 continues to be a candidate. But the next time a disk-discovery procedure runs, the candidate will be found invalid and the client will no longer have access the disk.

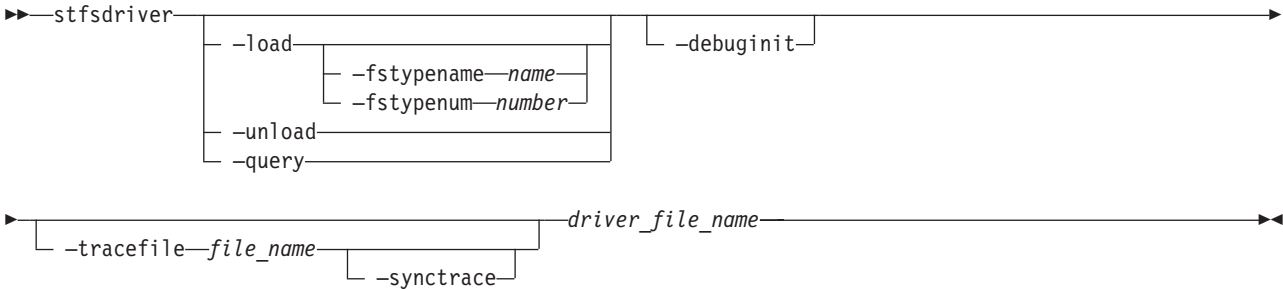
**Example**

**Query the disk-candidate list** The following example queries disk-candidate list and displays the status of each disk:

```
stfsdisk -query -kmname /usr/tank/client/bin/stfs
```

**stfsdriver**

Loads the file-system driver as a kernel extension.



**Parameters**

- load**  
Loads the kernel extension and create an instance of the file-system driver.
- unload**  
Unloads the kernel extension and destroys the instance of the file-system driver.



**-query**

Displays information about the kernel extension matching the specified criteria. For example, you can query the kernel extension ID to use in commands instead of the kernel module name.

**-fstypename** *name*

Specifies the name of the file-system type to use for the file-system-driver instance. This name relates to a specific file-system-type number. The file `/etc/vfs` maps the file-system-type name to the number.

If you do not specify a file-system-type name or number, the system defaults to the file-system-type named "sanfs". If there is no such type in the `/etc/vfs` file, the system defaults to the file-system-type number 20.

You will use this name to create the virtual client.

**-fstypenum** *number*

Identifies the number associated with the file-system type for the file-system-driver instance. All mount requests for a file system of this type are routed to this file-system-driver instance.

You would use this parameter only when you load multiple instances of the file-system driver on the same client system.

**Restriction:** Do not specify the number 1, 3, 16, or any already loaded file-system type number, or a number greater than 39.

**-debuginit**

Enables the file-system driver to issue diagnostic messages of the CONFIG class. Messages in this class are issued only during initialization.

If specified, the file-system driver does not issue diagnostic messages. You can turn the messages on after the file-system driver is running using the **stfsdebug** command.

**Note:** This parameter is intended for use only by trained service technicians.

**-tracefile** *file\_name*

Specifies that the file-system driver is to write diagnostic information to the specified file.

**Note:** The specified file must already exist.

**-synctrace**

Specifies that the file-system driver is to write diagnostic information to the specified trace file synchronously rather than using buffered writes.

*driver\_file\_name*

Specifies the name and location of the file-system driver that you want to load, unload, or query. The file name is typically "sanfs".

The file-system driver is loaded as a kernel extension. To identify the instance of the file-system driver, you identify the kernel extension. The kernel-extension name is the same as name and location of the file-system driver that was used to load the driver (for example, `/usr/tank/client/bin/sanfs`).

**Prerequisites**

You must have root privileges to use this command.

## Description

This command creates a file-system-driver instance by loading the file-system driver as a kernel extension. This command also unloads or queries the kernel extension.

After loading the file-system driver, you can use the **stfsclient** command to create a virtual client and then use the **stfsmount** command to mount the global namespace.

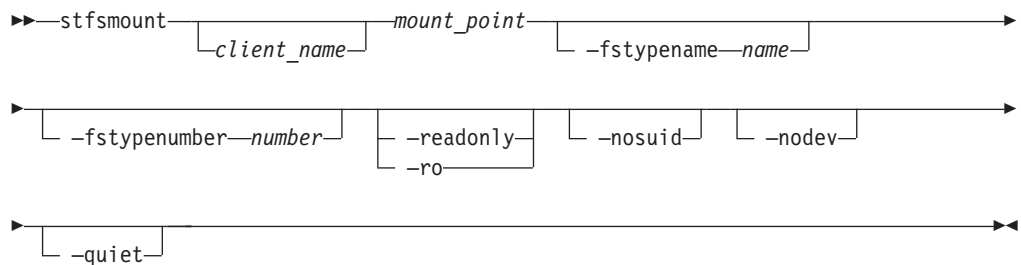
## Example

**Loads the file-system driver** The following example loads the file-system driver on a client for AIX:

```
stfsdriver -load /usr/tank/client/bin/sanfs
```

## stfsmount

Mounts the global namespace.



## Parameters

### *client\_name*

Identifies the unique name of the virtual client to which you want to mount the global namespace. The client must be up and running. The default client name is the host name of the client system.

### *mount\_point*

Specifies the directory associated with the global namespace image that you want to mount.

### *-fstypename name*

Specifies the name of the file-system type to use for the file-system-driver instance. This is the same name used to load the file-system driver.

This name relates to a specific file-system-type number. The file */etc/vfs* maps the file-system-type name to the number.

If you do not specify a file-system-type name or number, the system defaults to the file-system-type named "sanfs." If there is no such type in the */etc/vfs* file, the system defaults to the file-system-type number 20.

You would use this parameter only when you load multiple instances of the file-system driver on the same client system.

### *-fstypenumber number*

The number that identifies the file-system type for the file-system-driver instance. All mount requests for a file system of this type are routed to this file-system-driver instance.

**-readonly | -ro**

Sets the global namespace image to read only. If specified, an attempt to update data or metadata in the global namespace will fail, and an attempt to access a file-system object will not update its access-time attribute.

**-nosuid**

Disallows any invocation of the `setuid` or `setgid` commands from this file-system image.

**-nodev**

Disallows any attempts to open device nodes in this file-system image.

**-quiet**

Turns off informational messages for this command. This parameter does not affect error messages.

## Prerequisites

You must have root privileges to use this command.

## Description

This command creates an image of the global namespace on the client system by mounting a directory. The global namespace maintains a list of its directories that are available to the clients. When a client mounts a directory in the global namespace, that directory and its subdirectories become part of the client's directory hierarchy.

**Note:** This command is used in place of the **mount** command to mount the global namespace.

Before you can mount the global namespace, you must have a virtual client running on the client system. To create the virtual client, use the **stfsclient -create** command.

Remounting the global namespace image is not the same as unmounting the global namespace and then mounting it again. Rather, it changes the attributes of an existing global namespace image, such as changing from read-write to read-only mode. To remount the global namespace image or to see what global namespace images currently exist, use the **stfsmount** command.

To unmount the global namespace, use the **stfsumount** command.

## Example

**Mount the global namespace** The following example mounts the global namespace:

```
stfsmount mnt/SANFS_MOUNTPT -fstypename sanfs
```

## stfsstatus

Displays the version of the file-system driver for the specified virtual client for AIX.

►►—stfsstatus— -kmname—kernel\_ext\_name—◄◄

## Parameters

### **-kmname** *kernel\_ext\_name*

Identifies kernel-extension name of the file-system driver associated with the virtual client.

The file-system driver is loaded as a kernel extension. To identify the instance of the file-system drive, you identify the kernel extension. Each kernel extension has a name, but this name is not unique. This name is usually the file name of the object file from which you loaded the kernel extension (for example, /usr/tank/client/bin/stfs). To determine the kernel-extension name, use the **genkex | grep stfs** command.

## Prerequisites

You must have root privileges to use this command.

## Description

After issuing this command, if the client is running, the version of the file-system driver is displayed. If the file-system driver is not loaded, an error message is displayed stating that system could not determined the file-system driver instance.

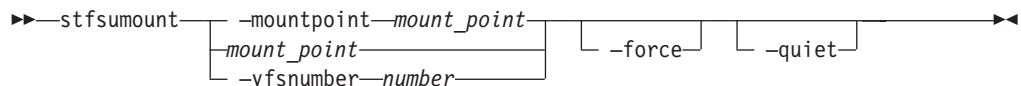
## Example

**Display the file-system-driver version** The following example displays the version of the file-system driver for the local client:

```
stfsstatus -kmname /usr/tank/client/bin/stfs
```

## stfsumount

Unmounts the global namespace.



## Parameters

### **-mountpoint** *mount\_point* | *mount\_point*

Specifies the directory associated with the global namespace image that you want to destroy. This must be the same directory that you specified to mount the global namespace.

If you created multiple global namespace images over the same directory (mount point), this command chooses the most recently created directory.

### **-vfsnumber** *number*

Identifies the virtual file-system (VFS) number associated with the global namespace image that you want to destroy.

In AIX, every global namespace image has a unique VFS number. The **stfsmount** command displays this number when it creates the global namespace image.

### **-force**

Unmounts the file system even if it is in use.

**-quiet**

Turns off informational messages for this command. This parameter does not affect error messages.

**Prerequisites**

You must have root privileges to use this command.

**Description**

This command destroys a global namespace image on the client system. It is used in place of the AIX **umount** command.

After you destroy all global namespace images that are linked to a specific virtual client, you can destroy the virtual client using the **stfsclient -destroy** command.

To see what global namespace images currently exist, use the AIX **mount** command with no parameters.

**Example**

**Unmount the global namespace** The following example unmounts the global namespace on the local client:

```
stfsumount -mountpoint sanfs/cnt1
```

**Linux-client commands**

This topic provides a brief description for each Linux-client command.

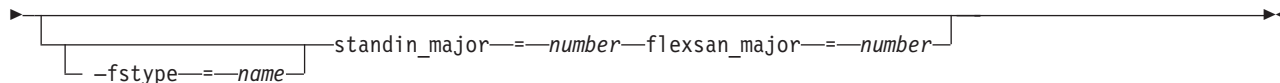
**Note:** You must have root privileges to use these commands.

Command	Description
<b>Migration</b>	
"migratedata" on page 264	Migrates data to SAN File System.
<b>Virtual client setup and removal</b>	
"insmod"	Loads the file-system driver as a kernel module.
"rmmod" on page 279	Unloads the file-system driver as a kernel module.
"rmstclient" on page 267	Unmounts the global namespace, removes the SAN File System client, and unloads the file-system driver.
"setupstclient" on page 267	Configures and starts a client.
"stfsclient" on page 280	Creates or destroys a virtual client.
"stfsmount" on page 282	Mounts the global namespace.

**insmod**

Loads the file-system driver as a kernel extension.

►►—insmod—*driver\_file\_name*—————►



## Parameters

*driver\_file\_name*

Specifies the name and location of the file-system driver that you want to load. The file name is typically **stfs.o**.

**fstype=** *name*

Specifies the name of the file-system type. The file system must be of the STFS type, but you can choose any name for that type when you load the file-system driver on the client system. Use the same file-system type name when you create or destroy the client. The default name is **stfs**.

Use this parameter when you load multiple instances of the file-system driver on the same client system. The file-system type name connects the file-system driver instance with a global namespace image.

**standin** **major=** *number*

Specifies the major device number for SAN File System. The default is 13.

flexsan major= *number*

Specifies the major device number for the flexible SAN. The default is 15.

### Prerequisites

You must have root privileges to use this command.

### Description

This command creates a file-system-driver instance by loading the file-system driver as a kernel extension. After loading the file-system driver, you can use the **stfsclient** command to create a virtual client and then use the **stfsmount** command to mount the global namespace.

### Example

**Load the file-system driver** The following example loads the file-system driver on a client for Linux:

```
insmod usr/tank/client/bin/stfs.o
```

**rmmod**

Unloads the kernel extension and destroys the instance of the file-system driver.



### Prerequisites

You must have root privileges to use this command.

## Description

This command unloads the kernel extension and destroys the instance of the file-system driver. Use the **stfsclient** command to destroy all instances of the virtual client before unloading the module.

## Example

**Unload the file-system driver** The following example unloads the file-system driver on a client for Linux:

```
rmmod stfs
```

## stfsclient

Creates or destroys a virtual client.

```
►► stfsclient -create [client_name] [server_name] [server_IP_address] [:-port] [fstype-name]
► [-quiet] [-nettype tcp | udp] [devices = [dir = [directory] | pat = [/dev/sd*[a-z] | pattern]]
► -converter-8859-1
```

or

```
►► stfsclient -destroy [client_name] [fstype-name]
```

## Parameters

### -create

Creates a new virtual client.

### -destroy

Destroys an existing virtual client.

### *client\_name*

Identifies the unique name of the virtual client that you want to create or destroy. The default client name is the host name of the client system.

### *server\_name*

Specifies the host name of a metadata server in the SAN File System. The metadata server that you specify informs the global namespace image of all other metadata servers.

This parameter is not required if this is not the first mount for a particular virtual client.

### *server\_IP\_address*

Specifies the IP address, in dotted decimal notation, of a metadata server in the SAN File System.

### *port*

Specifies the port number of the specified metadata server. The default is 1700.

**-fstype** *name*

Specifies the name of the file-system type. The file system must be of the STFS type, but you can choose any name for that type when you load the file-system driver on the client system. You must use the same name that you used to load the file-system driver. The default name is **stfs**.

Use this parameter when you load multiple instances of the file-system driver on the same client system. The file-system type name connects the file-system driver instance with a global namespace image.

**-devices**

Determines which devices, also called disks or logical unit numbers (LUNs), that the virtual client considers as SAN File System volumes. The default is **-devices=pat=/dev/sd\*[a-z]**, where \* represents any alphabetic characters (a-z).

In addition to creating the virtual client, this command discovers which disks, or candidates, are available to the virtual client as volumes and transmits the candidate list to the virtual client. The **-devices** parameter controls the candidates list.

**dir=***directory*

The candidate list is made up of those devices that have device special files in the specified directory (for example:  
**-devices=dir=/dev/stfsdisk**).

The easiest way to mount the global namespace is to specify **-devices=pat=/dev/sd\*[a-z]**, where \* represents any alphanumeric character (a-z, A-Z, 0-9). Specifying the parameter in this way causes the client to look at every SCSI-disk-like device in the system. Whatever looks like a SAN File System disk is accessed when the metadata server refers to that disk's SAN File System disk identifier.

If you want the client to be more selective about what disks it considers available, you can create a **/dev/stfsdisk** directory, put device-special files (or symbolic links) for your candidates in it, and use **-devices=dir=/dev/stfsdisk**.

**pat=***pattern*

The candidate list is made up of those devices that have device-special files whose file specifications match the specified pattern. You can use \* wildcards in the last (filename) component but not in the directory components (for example, **-devices=pat=/dev/sd\*[a-z]**).

**none** The candidate list is empty. Use this value when you want to establish the candidate list with a separate command, perhaps using a selection method more sophisticated than the **stfsclient** command offers.

**-quiet**

Turns off informational messages for this command. This parameter does not affect error messages.

**-nettype**

Specifies the protocol to be used between the client and server (UDP or TCP). All clients must use the same protocol. The default is TCP.

**Prerequisites**

You must have root privileges to use this command.



## Description

This command creates or destroys a virtual client. A *virtual client* is an entity that communicates with a metadata server and, indirectly, with other SAN File System clients. In this release, only one virtual client can be used per client machine. The terms *virtual client* and *client* can be used interchangeably.

A virtual client is associated with exactly one SAN File System. There is one file cache and one set of disk candidates per virtual client. Each virtual client that is running on the same system is as separate as if it were running on a different system. They share nothing except the file-system drive code that they execute.

A SAN File System virtual client is uniquely identified in the context of its file-system driver, and in the context of its SAN File System, by its client name.

To use the files in a global namespace, the virtual client must have a global namespace image. Creating a global namespace image makes the directory structure in the global namespace appear in the client's file structure. To create a global namespace image, use the **stfsmount** command.

A client can access and create data that is stored in a global namespace. Each virtual client can access data on multiple images in the same global namespace.

The client considers a file to be one file even if it appears with two different file names in two different global namespace images.

For Linux clients, to view the existing SAN File System virtual clients, look in the `proc/fs` file system, in the directory named after the file-system type (usually **stfs**). In that directory, there is a subdirectory for each virtual client. The name of the subdirectory is the same as the client name.

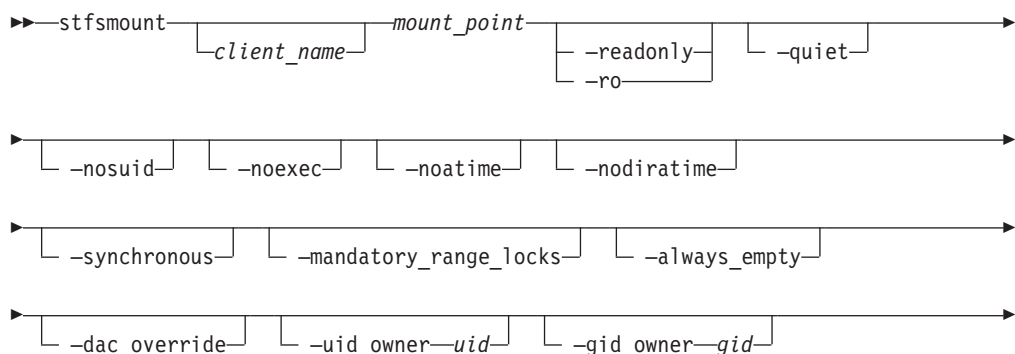
## Example

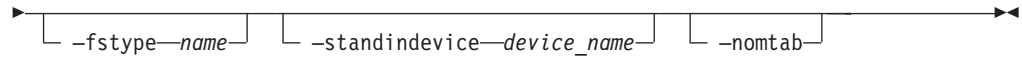
**Create a virtual client** The following example creates a virtual client with the candidates as volumes that have device-special files, with file specifications that match the pattern `sd*[a-z]`:

```
stfsclient -create MDS1:1700 -devices=pat=/dev/sd*[a-z] -converter 8859-1
```

## stfsmount

Mounts the global namespace.





## Parameters

### *client\_name*

Identifies the unique name of the virtual client to which you want to mount the global namespace. The client must be up and running. The default client name is the host name of the client system.

### *mount\_point*

Specifies the directory associated with the global namespace image that you want to mount.

### **-readonly | -ro**

Sets the global namespace image to read only. If specified, an attempt to update data or metadata in the global namespace will fail, and an attempt to access a file-system object will not update its access-time attribute.

### **-quiet**

Turns off informational messages for this command. This parameter does not affect error messages.

### **-nosuid**

Ignores the set user ID and set group ID flags on files that are accessed in the global namespace image. You cannot get special privileges by executing these files.

### **-noexec**

Ignores the exec flag on files accessed in the global namespace image, except to indicate its existence. You cannot execute files.

### **-noatime**

Specifies not to update the access-time attribute when a file-system object is accessed in the global namespace image.

**Note:** The virtual client does not update the access time of a file-system object when it accesses its attributes. It only updates the access time when it accesses its contents.

### **-nodiratime**

Specifies not to update the access-time attribute when a directory is accessed in the global namespace image.

**Note:** The virtual client does not update the access time of a directory every time it is accessed. It updates the access time when you read the directory, but not when you look up a specific name in it.

### **-synchronous**

Specifies not to return system calls that update file data or metadata in this global namespace image until after the updates have been committed to storage.

Without this parameter, system calls return as soon as the changes are stored in kernel-cache memory, and the changes are committed to storage some time later (automatically or by user request).

### **-mandatory\_range\_locks**

Allows mandatory range locking to be available in the global namespace image.

Without this parameter, all range locks set or seen through the global namespace image are advisory, as defined by POSIX. This means that you can lock a range of a file, but no one has to respect that lock, and you do not have to respect anyone else's locks.

With the parameter, some range locks can be mandatory to an extent. Range locks do not have an advisory or mandatory state. But when you access a range of a file that is locked, Linux treats the lock as mandatory or advisory depending on whether the file image that you access is in mandatory or advisory range lock mode. A file image is in mandatory-range-lock mode if it was created while the global namespace image had mandatory range locks allowed and the file had the set group ID flag on and the group execute flag off.

Mandatory range locks are considered mandatory only with respect to a particular global namespace image.

A client does not know whether another client considers a lock mandatory or advisory. So, if you access a file using multiple processes on the same system, through the same global namespace image, mandatory locks are fully mandatory. But another client, or even a process accessing the same client from a different global namespace image, is free to consider the locks advisory.

When the owning group of a file changes, its set group ID flag gets turned off, and range locks are no longer mandatory. This is normal behavior for the set group ID flag, but in many Linux file-system types, it is suspended for the special case of the set group ID flag that means mandatory range lock. SAN File System does make that special exception because the flag does not have the same meaning on non-Linux client. Note, that as with any Linux mandatory range lock, they do not restrict access to the file using a private mmap.

You can change the mandatory-range-locks-allowed status of a global namespace image by remounting.

The mandatory-range-lock status takes effect when a file is opened. Changing the mandatory-range-lock attribute for the global namespace image after you open the file does not change the file's status. For example, if you have a file open with mandatory range locks allowed, and then you disallow mandatory range locks, all locks on that file, including subsequent ones, are still mandatory until you close the file.

#### **-always\_empty**

The Linux **statfs** command retrieves the approximate number of available blocks and free blocks in the global namespace. The number of *available blocks* is the total number of blocks in all storage pools presently in all the clusters that are reachable through the global namespace image. The number of free blocks is the number of blocks in all the clusters that are in partitions that are not assigned to any fileset.

**Note:** Based on these numbers, an application designed for a simpler file system might conclude that there is no space available to store new data when in fact there is plenty of space within partitions that are already assigned to the fileset, or that there is a large pool of storage not presently attached to a cluster but ready to be attached when demanded. If specified, the **statfs** command returns the largest number it can for the number of available blocks and returns that number minus one for the number of free blocks. If the number of blocks is too large for the **statfs** command to return, the command behaves as if you

specified the `-always_empty` parameter. This is the case when you have more than 16 TB of space in the SAN File System.

**-dac\_override**

Enables capability to override regular permissions on a file, regardless of root client status.

**-uidowner** *uid*

Sets the owner user ID for each file accessed in the global namespace image. The owner user ID stored in the file system is ignored.

If not specified, the global namespace image uses the real owner user ID stored in the file system.

**Note:** You cannot set the owner user ID in the global namespace. An attempt to set it to user ID does succeed, but the global namespace does not actually get changed. An attempt to set it to anything else fails.

**-gidowner** *gid*

Sets the owner group ID for each file accessed in the global namespace image.

**-fstype** *name*

Specifies the name of the file-system type to use for the file-system-driver instance. This is the same name used to load the file-system driver.

If you do not specify a file-system-type name, the system defaults to the file-system-type named "stfs".

Use this parameter only when you load multiple instances of the file-system driver on the same client system.

**-standindevice** *device\_name*

Specifies the device-specific file name of the standin block device for the global namespace image. If you specify a standin block device that is already being used for an existing global namespace image, this command does not create a new global namespace image. Instead, it mounts the existing global namespace image again, without disturbing the existing mounts.

In order to make Linux NFS export SAN File System files, the file-system type presents itself to Linux as a device-based file-system type, as opposed to a network file-system, such as NFS. But because the global namespace does not actually live on a device (it lives on a collection of devices and a metadata server), a block device must stand in for the global namespace wherever it makes a difference. Accordingly, every global namespace image is associated with a block device, known as the *standin block device*. The file-system driver contains a block-device driver and constitutes a set of very simple block devices whose only job is to be standin block devices. They cannot be used for anything. If you exports SAN File System files from NFS, make sure you do not use the same standin block device for two different SAN File Systems (at different times).

**Note:** You must specify one of the file-system-driver instance's standin block devices. You can see what these are in the standin proc file for the file-system type.

Note that the device is identified with the global namespace image by device number, not by the device-specific file name. You can delete the device-specific file after the **stfsmount** command completes.

The value of this parameter is what shows up as the file-system identifier in the Linux mount table listing in `/proc/mounts`.

You can create a block-device-specific file using the Linux **mknod** command.

If not specified, the **stfsmount** command uses an unused stdin block device. It creates a temporary block-device-specific file for it and adds the name of that temporary file to the Linux mount table. It creates the file in the directory defined by the TMPDIR environment variable, or /tmp if TMPDIR is not defined. It uses a lock file to prevent two processes that are simultaneously running the **stfsmount** command from choosing the same unused stdin block device. It creates that file in the same temporary directory as the block-device-specific file.

#### **-nomtab**

Specifies not to record the mount in the /etc/mtab file.

If not specified, the **stfsmount** command adds an entry to the /etc/mtab file describing the global namespace image if the mount is successful. It then locks the file using the /etc/mtab~ lock file while it attempts the mount and updates /etc/mtab. If specified, this command does not attempt to lock or update /etc/mtab.

Note that /etc/mtab is obsolete and would only need to be updated if it might be referenced by existing applications. /proc/mounts contains more reliable information, straight from the kernel itself.

#### **Prerequisites**

You must have root privileges to use this command.

#### **Description**

This command creates an image of the global namespace on the client system by mounting a directory. The global namespace maintains a list of its directories that are available to the clients. When a client mounts a directory in the global namespace, that directory and its subdirectories become part of the client's directory hierarchy.

**Note:** This command is used in place of the **mount** command to mount the global namespace.

Before you can mount the global namespace, you must have a virtual client running on the client system. To create the virtual client, use the **stfscient -create** command.

Remounting the global namespace image is not the same as unmounting the global namespace and then mounting it again. Rather, it changes the attributes of an existing global namespace image, such as changing from read-write to read-only mode. To remount the global namespace image you can use the Linux **mount** command.

To unmount the global namespace, use the Linux **umount** command.

#### **Example**

**Mount the global namespace** The following example mounts the global namespace:

```
stfsmount mnt/SANFS_MOUNTPT -fstype sanfs
```

## Solaris-client commands

This topic provides a brief description for each Solaris-client command.

**Note:** You must have root privileges to use these commands.

Command	Description
<b>Migration</b>	
"migratedata" on page 264	Migrates data to SAN File System.
<b>Status</b>	
"sanfs_ctl stats" on page 291	Displays statistics about the client.
<b>Volumes and LUNs</b>	
"sanfsd" on page 294	Starts the sanfsd program that responds to data LUN operation requests to the client. This program starts automatically after a successful mount. All data LUN operations are initiated from and coordinated by the metadata servers in the cluster but are actually performed by one or more clients.
"sanfs_ctl disk" on page 290	Adds, removes, and lists disks. This command is not necessary when the sanfsd program is running.
<b>Virtual client setup</b>	
"labellun"	Creates a Solaris label on a disk.
"mount" on page 288	Mounts the global namespace.
"rmstclient" on page 267	Removes the virtual client for Solaris.
"setupstclient" on page 267	Configures and starts a client.
<b>Problem determination</b>	
"sanfs_ctl trace" on page 293	Enables and disables tracing.

### labellun

Creates a Solaris label on a disk.

►► labellun ☐\_f ☐\_s diskname ◀◀

#### Parameters

- f Forces the system to create a Solaris label on the disk even if a label already exists.
- s Runs silently, turns off confirmation messages for this command.

*diskname*

Identifies the disk on which you want a Solaris label.

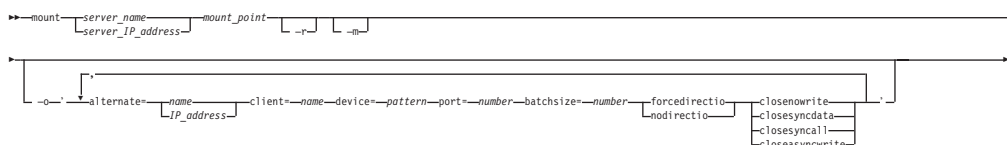
#### Prerequisites

You must have root privileges to use this command.

**Label a disk** The following example labels /dev/dsk/c2t9d5s2 as a Solaris disk:

```
label1un /dev/dsk/c2t9d5s2
```

Mounts a client instance of the SAN File System.



## server name

Specifies the host name of a metadata server in the SAN File System. The metadata server that you specify informs the global namespace image of all other metadata servers.

Specifies the IP address, in dotted decimal notation, of a metadata server in the SAN File System.

Specifies the directory on which the client instance is mounted.

- r Sets the global namespace image to read only. If specified, an attempt to update data or metadata in the global namespace fails, and accessing a file-system object does not update its access-time attribute.

Specifies not to record the mount in the `/etc/mstab` file.

If `-m` is not specified, the **mount** command adds an entry to the `/etc/mtab` file describing the global namespace image if the mount is successful. It then locks the file using the `/etc/mtab~` lock file while it attempts the mount and updates `/etc/mtab`. If specified, this command does not attempt to lock or update `/etc/mtab`.

Note that `/etc/mtab` is obsolete and would only need to be updated if it might be referred to by existing applications. `/proc/mounts` contains more reliable information, straight from the kernel itself.

- o Specifies options that are specific to SAN File System. Enclose the options in single quotes and separate the options with a comma and no space.

Specifies the host name or the IP address, in dotted decimal notation, of an alternate metadata server in the SAN File System if the metadata server specified by the `--server` parameter does not respond.

Identifies the unique name of the virtual client to which you want to mount the global namespace. The client must be up and running. The default client name is the host name of the client system.

Specifies the devices (also called disks or LUNs) that are usable for I/O operations. The default is `/dev/dsk/c*t*d*s2`.

**port=** *number*

Specifies the port number of the specified metadata server to which the client should connect. The default is 1700.

**batchsize=** *number*

Specifies the maximum number of SAN File System messages that can be batched together in a single network request to the metadata server.

**forcedirectio**

Causes all file access on this mount instance to use direct I/O.

**nodirectio**

Inhibits the use of direct I/O on this mount instance.

**closerowrite**

Specifies that when a file is closed, the system not attempt to write any unwritten data back to disk at that time. Solaris flushing mechanisms or pageout operations will force the data to disk.

**closesyncdata**

Specifies that only file data, not metadata, be written back to disk when a file is closed.

**closesyncall**

Specifies that file data and metadata be written back to disk when a file is closed.

**closeasyncwrite**

Specifies that file data and metadata be asynchronously written back to disk when a file is closed. The operations to write the data to disk start when the file is closed. However, the close() call does not wait for the operations to complete; it returns prior to all data being written to disk.

## Prerequisites

You must have root privileges to use this command.

## Description

This command creates an image of the global namespace on the client system by mounting a directory. The global namespace maintains a list of its directories that are available to the clients. When a client mounts a directory in the global namespace, that directory and its subdirectories become part of the client's directory hierarchy.

To unmount the global namespace, use the Solaris **umount** command.

## Example

**Mount the global namespace** The following example mounts the global namespace:

```
mount MDS1 /mnt/sanfs -o 'port=1800,alternate=MDS2,alternate=MDS3,device=/dev/dsk/c1t2d*s2'
```



## rmstclient

Removes the virtual client for Solaris.



### Parameters

#### **-prompt**

Prompts for required parameters, using values from the configuration file, if available.

#### **-noprompt**

Runs silently, using parameters from the configuration file (`/usr/tank/client/config/stclient.conf`). If a required parameter is not available, the command exists with an error.

### Prerequisites

You must have root privileges to use this command.

You must unmount the SAN File System before invoking this command.

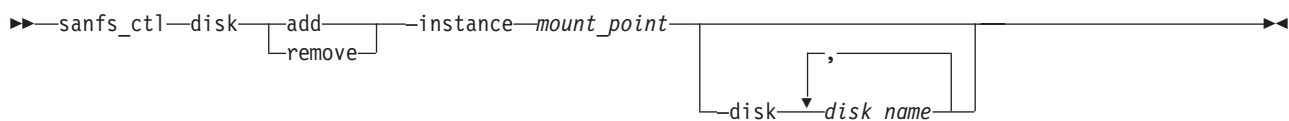
### Example

**Remove a client for Solaris** The following example removes the local SAN File System client for Solaris without prompting:

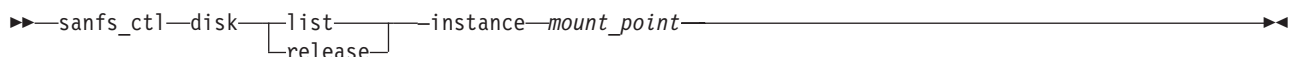
```
rmstclient -noprompt
```

## sanfs\_ctl disk

Controls the SAN File System volumes (disks) that a client can access and lists accessible disks. The need to use this command is greatly reduced when the `sanfsd` program is running.



or



### Parameters

#### **add**

Adds the specified disk file name to the list of disks that the Solaris client uses for I/O operations.

#### **remove**

Removes the disk from the list of disks that the Solaris client uses for I/O operations.

## list

Lists all the disks that are accessible from the given instance.

## -release

Removes all the disks that are accessible from the given instance.

## -instance *mount\_point*

Specifies the directory on which the client instance is mounted.

## -disk *disk\_name*

Specifies one or more names of the disks to add or remove. When you specify more than one name, separate the names with a comma and no space (for example, -disk /dev/dsk/disk1,/dev/dsk/disk2).

## Prerequisites

You must have root privileges to use this command.

## Description

The **sanfs\_ctl disk** command enables manual control over the disks used by the SAN FS Solaris client when it performs I/O operations. When the sanfsd program is running, there is little need to use this command.

## Example

**List accessible disks** The following example lists all the disks that are accessible for the local client for Solaris that is mounted on /mnt/sanfs:

```
sanfs_ctl disk list -instance /mnt/sanfs
```

## sanfs\_ctl stats

Displays statistics about the Solaris client.

```
▶▶ sanfs_ctl stats get instance mount_point [data[,data_type]] [short]
```

or

```
▶▶ sanfs_ctl stats list
```

or

```
▶▶ sanfs_ctl stats reset instance mount_point
```

or

```
▶▶ sanfs_ctl stats monitor instance mount_point [-host host_name -p port]
```

```
▶ -interval interval
```

## Parameters

### **get**

Retrieves performance information for a single, multiple, or all data items.

### **list**

Lists the names of all statistics that can be gathered.

### **reset**

Resets the statistics for all the data types.

### **monitor**

Specifies that the utility continuously generate statistics.

### **-instance** *mount\_point*

Specifies the directory on which the client instance is mounted.

### **-data** *data\_type*

Identifies the data types for which the utility gathers statistics. When you specify more than one data type, separate the data types with a comma and no space. The data types can be the following values:

- **tm** - Transaction Manager (TM) statistics per client related to TM data structures, including the number of messages sent and received (per message type), the maximum lengths and average lengths of the transaction queue, and the number of transactions, messages, and leases lost.
- **net** - Network statistics per metadata server with which this client has a lease.
- **mc** - Metadata-cache statistics.
- **ho** - The states of all objects in the object hash table, including object ID, flags, and attributes.
- **lo** - The states of all objects in the object LRU list, including object ID, flags, and attributes.
- **ls** - Mutex and latch statistics of metadata cache objects. Statistics include the number of attempts, wait time, and hold time.
- **le** - Mutex and latch state of all objects in the object hash table, including current holder, hold state, and waiters.

### **-short**

Displays the minimal information for the specified data items.

### **-host** *host\_name* **-port** *port\_number*

The statistics are sent as a string to the specified *host\_name* on the specified UDP *port*. If no host or port is specified, the statistics are sent to stdout.

### **-interval** *interval*

Specifies the *interval* (in seconds) at which the utility generates statistics when the monitor parameter is specified.

## Prerequisites

You must have root privileges to use this command.

## Description

The **sanfs\_ctl stats get** command enables you to retrieve statistics about specific or all data types. You can use the **sanfs\_ctl stats list** command to see a list of the available data types. The **sanfs\_ctl stats reset** command enables you to reset the gathering of these statistics.

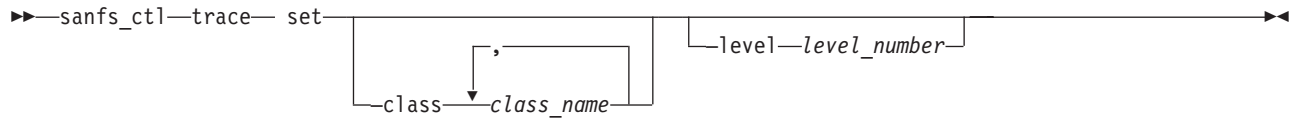
## Example

**Displays statistics from a client for Solaris** The following example displays transaction-manager statistics and metadata-cache statistics for the local client for Solaris that is mounted on /mnt/sanfs:

```
sanfs_ctl stats get -instance /mnt/sanfs -data tm,mc
```

## sanfs\_ctl trace

Enables and disables tracing.



or



## Parameters

### set

Sets a the tracing level for one or more specified trace classes or all trace classes.

### list

Lists trace status.

### -class *class\_name*

Identifies one or more classes for which to set the trap. When you specify more than one class name, separate the names with a comma and no space. A maximum of 20 class names can be specified for each `-class` parameter.

### -level *level\_number*

Specifies the tracing level. Possible values for the level number are 0 through 5. Setting the level to 0 turns off tracing for the specified class.

## Prerequisites

You must have root privileges to use this command.

## Description

The trace records are written to a buffer in memory and sent to the system log file.

This command enables you to group class names that you want to set on or off. If you do not specify a class, all trace classes are affected by the specified trace level.

## Example

**List trace levels** The following example lists traces with no tracing enabled:

```
sanfs_ctl trace list
```

CSM Trace Class	Value
=====	=====
CSM:LEASE	0
CSM:XMTRCV	0

CSM:OBJECT	0
CSM:CACHE	0
CSM:MC_SES	0
CSM:CSM_SES	0
CSM:MC_DATA	0
CSM:CSMDATA	0
CSM:MC_RNGE	0
CSM:CSMRNGE	0
CSM:ALLOC	0
CSM:OBJATTR	0
CSM:SOCKET	0

ULD Trace Class	Value
=====	=====
ULD:FLOW	0
ULD:CONFIG	0
ULD:CTRLIFC	0
ULD:MOUNT	0
ULD:DISK	0
ULD:PAGER	0
ULD:VFSOP	0
ULD:VNODEOP	0
ULD:RNGLOC	0
ULD:CAPTURE	0

## sanfsd

Starts the **sanfsd** program on a Solaris client. This program responds to data LUN operation requests from metadata servers. The program is started automatically when the **sanfs\_mount** command succeeds. The **sanfsd** command enables you to restart the program if it stops.

►►—sanfsd—instance—*mount\_point*—device—*pattern*—————◄◄

## Parameters

—**instance** *mount\_point*

Specifies the directory on which the client instance is mounted.

—**device** *pattern*

Determines which devices (also called disks or LUNs) that the client considers as SAN File System volumes. An example device pattern is `/dev/dsk/c*t*d*s2`. Enclose the pattern in single quotes so the shell does not expand the asterisk characters.

## Prerequisites

You must have root privileges to use this command.

## Description

The **sanfsd** program starts automatically when the **sanfs\_mount** command successfully runs a mount invocation. A **sanfsd** program runs for each SAN File System mount. The **sanfsd** program stops when its accompanying mount point is unmounted. If the program is not running for a particular mount, the client cannot respond to data LUN operations until the program is restarted.

## Example

**Start the **sanfsd** program.** The following example starts the **sanfsd** program for the local client for Solaris that is mounted on `/mnt/sanfs`. :

```
sanfsd -instance /mnt/sanfs -device '/dev/dsk/c*t*d*s2'
```

## Windows-client command

The topic provides a brief description for the Windows-client command.

**Note:** You must have Administrator privileges on the Windows client to use these commands.

Command	Description
<b>Migration</b>	
"migratedata" on page 264	Migrates data to SAN File System.

---

## SAN File System installation commands

This topic provides an overview of the SAN File System installation commands.

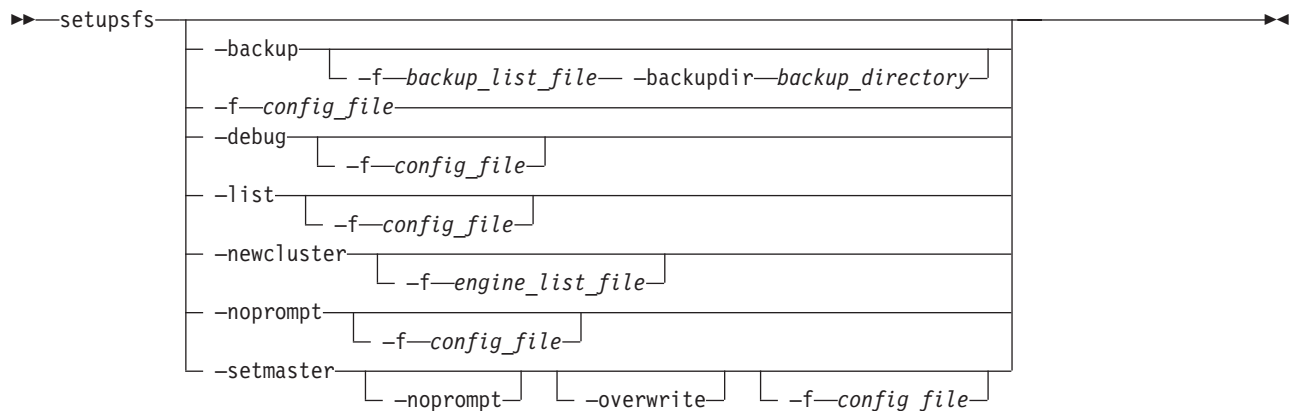
The SAN File System installation commands include:

- **Setupfs.** Used to configure and start a SAN File System metadata server.
- **Setupstclient.** Used to configure and start a SAN File System client.
- **Tmvt.** Used to verify that the hardware and software prerequisites for the SAN File System are installed.

In addition, information about the administrative commands and all client commands are provided in the *SAN File System Administrator's Guide*.

### setupfs

Configures and starts a SAN File System metadata server



#### Parameters

##### **-backup** *backup\_file*

Creates a backup (tar archive) containing all metadata server configuration information. The list of files to be backed up up are specified in /usr/tank/admin/config/backup.list. You can specify an alternate backup file list using the -f parameter.

By default the archive is stored as /usr/tank/server/DR/DRfiles-hostname-date.tar.gz

To restore a metadata server from a backup that was created using the `–backup` parameter, extract the tar archive in the root directory. For example:

```
tar -xzf /usr/tank/server/DR/DRfiles-hostname-date.tar.gz
```

Then, run **setupsfs** without parameters (or with only `–noprompt` parameter) to restart the metadata server.

**Attention:** Do not use the `–setmaster` parameter when restoring from a backup.

**–backupdir** *directory\_path*

Specifies an alternate directory, in which to create the archive.

**–debug**

Provides extra parameters with defaults. The most common use of this parameter is when you are using ActiveDirectory as your LDAP server.

**–f** *file*

Specifies the name of a file that contains configuration information, list of engines, or tar archive, depending on the parameters specified before the `–f` parameter.

**–list**

List all parameters and the corresponding values found in the configuration file.

**–noprompt**

Runs the **setupsfs** command without prompting you for information. The configuration file is expected to exist. If the configuration file does not exist or if a required value is missing or invalid, **setupsfs** exits with an error.

**–newcluster** *engine\_list*

Adds one or more subordinate metadata servers to the cluster. Use this parameter only on the master metadata server.

The *engine\_list* identifies the subordinate nodes to be added. All subordinate metadata servers must be running at the time that **setupsfs –newcluster** is started.

**–overwrite**

Initialize the master metadata server and system disks, regardless of whether they already contain cluster information.

**Attention:** Using this parameter will destroy all data stored in the system storage pool.

**–setmaster**

Configures the metadata server as the master metadata server. If the given master disk already contains cluster information, this information is used when starting the metadata server.

## Description

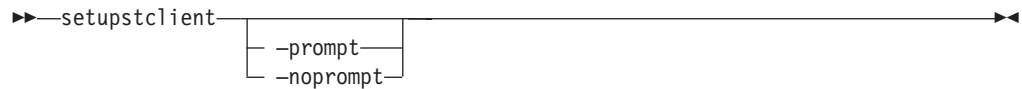
The **setupsfs** command is used to configure, start, or restart a metadata server. It can also be used to display the current configuration values and back up the configuration in case the server needs to be restored at a later date.

If you do not specify any parameters, **setupsfs** prompts you for the configuration values required to set up a subordinate metadata server. The **setupsfs** command maintains the values in a configuration file in `parameter=value` format. The

configuration file is /usr/tank/admin/config/tank.properties. If a configuration file already exists, its values will be presented as the suggested defaults when the prompt is displayed. Otherwise, the manufacturing defaults are presented.

## setupstclient

Configures and starts SAN File System clients.



### Parameters

#### **-prompt**

Forces the **setupstclient** command to prompt for all configuration values.

#### **-noprompt**

Runs silently, using parameters from the configuration file. If the configuration file does not exist or if a required parameter is not available or invalid, the command exits with an error.

### Prerequisites

You must have root privileges to use this command.

### Description

This command configures and starts, or restarts a SAN File System client.

If you do not specify a parameter, this command runs silently using values from the configuration file as defaults. It only prompts for any required information, if a configuration file does not exist or if a value in an existing configuration file is not valid.

This command maintains any values given by the user in a configuration file in parameter=value format. The default configuration file is /usr/tank/client/config/stclient.conf.

Specify the **-prompt** parameter to force the command to prompt for all configuration values. In this case, if a configuration file exists, the command presents the value from the configuration file as the suggested default when the command displays a prompt. If a configuration file does not exist, the command presents the manufacturing default as the suggested default.

If you specify the **-noprompt** parameter, the command expects the configuration file to exist. If the file does not contain valid values, the command exits with an error.

### Example

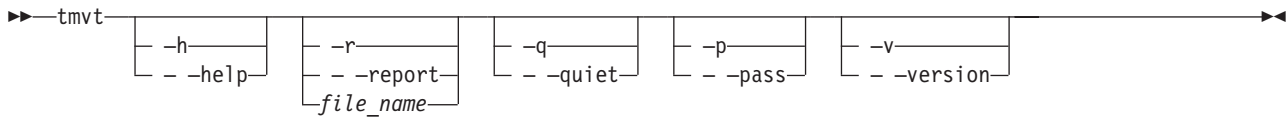
**Setup a client** The following example configures and starts SAN File System clients:

**setupstclient**



## tmvt

Validates that the existing hardware and software on a metadata server meet all installation requirements for the SAN File System.



### Parameters

#### -? | -h | --help

Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.

#### -r | --report *file\_name*

The file name into which the report is written. The default is to write the report to the standard output stream, stdout.

#### -q | --quiet

Run in quiet mode. Does not display any output.

#### -p | --pass

Force the return code from tmvt processing to be zero (hardware and software passes).

#### -v | --version

Output the version number of the **tmvt** command and the product release that it is validating.

### Description

**Note:** Exceptions are always written to stderr and tmvt analysis is written (by default) to stdout. If you specify a report file using the -report parameter, tmvt analysis is written to that file. Exceptions are written to the file and to stdout.

### Example

The following example validates the hardware and software for a metadata server:

**tmvt**

The following example shows a sample of a partial listing from the file.

Hardware Components (15)			
Item Name		Current	Recipe
Passed Hardware Component Checks (15)			
Memory (Megabytes)		4040	4000
Disk space in /var (Megabytes)		16472	4096
TCP/IP		enabled	enabled
Ethernet controller		Intel 82546EB	Intel 82546EB
Machine Type/Model		867061X	867061X
Machine BIOS Level		1.1.5	1.1.5
Machine BIOS Build		GEE148AUS	GEE152AUS
FC HBA Manufacturer		QLogic	QLogic
FC HBA Model		QLA2342	QLA2342
FC HBA BIOS/Firmware Version		3.02.16	3.02.16
FC HBA Driver Version		6.06.60	6.06.60

Remote Supervisor Adapter 2	present	present
Remote Supervisor Adapter 2 Firmware	GEB833A	GEB833A
Remote Supervisor Adapter 2 Driver	1.08	1.08
RS485 Service Processor	enabled	enabled

#### Software Components (794)

Item Name	Current	Recipe
Correct Software Packages (794)		
zsh	4.0.6-30	4.0.6-30
zoo	2.10-602	2.10-602
zlib-devel	1.1.4-51	1.1.4-51
zlib	1.1.4-51	1.1.4-51
zip	2.3-490	2.3-490
zebra	0.93b-74	0.93b-74
ypserv	2.9.91-27	2.9.91-27
ypbind	1.12-55	1.12-55
yp-tools	2.7-60	2.7-60

## Service commands and utilities

### Service commands

### Administrative commands

The following table provides a brief description and role for each command in the Administrative CLI that is intended for use only by trained service personnel.

Command	Description	Role	Environment
<b>Administrative server</b>			
“startCimom” on page 307	Starts the administrative agent.	root	shell
“stopCimom” on page 312	Stops the administrative agent.	root	shell
<b>Cluster</b>			
“mktruststore” on page 302	Creates a truststore that is shared by the administrative infrastructure to certify secure connections.	n/a	shell
“tank extractbootrecord” on page 313	Extracts a local Tank.Bootstrap file from a valid master volume.	Administrator	shell
“tank lscluster” on page 314	Displays the cluster definition from a system master volume when the metadata servers are not running.	Administrator	shell
“tank lsdisklabel” on page 317	Displays the a SAN File System product label for a specified device.	Administrator	shell

Command	Description	Role	Environment
“tank lsversion” on page 316	Displays the version control information from a system master disk.	Administrator	shell
“tank resetcluster” on page 319	Erases the static cluster definition contained on the system master volume.	Administrator	shell
“tank resetversion” on page 319	Resets the version-control information on a system master disk.	Administrator	shell
<b>Engine</b>			
“obdc” on page 303	Collects information for the first-failure data capture of SAN File System problems from the engine on which the metadata server resides.	Operator, Administrator	shell
<b>Legacy CLI</b>			
“legacy” on page 160	Runs commands in the legacy CLI.	Administrator	shell
<b>SAN File System console</b>			
“disableConsoleTrace” on page 301	Disables tracing and logging for the SAN File System console.	root	shell
“enableConsoleTrace” on page 301	Enables tracing and logging for the SAN File System console.	root	shell
“startConsole” on page 308	Starts the SAN File System console Web server.	root	shell
“stopConsole” on page 313	Stops the SAN File System console Web server.	root	shell

## Client commands

The following table provides a brief description and role for each client command that is intended for use only by trained service personnel.

Command	Description	Role
<b>Diagnostics</b>		
“obdc” on page 303	Collects information for the first-failure data capture of SAN File System problems from the client machine.	root (AIX) or Administrator (Windows)
“stfsdebug” on page 308	(AIX only) Enables or disables the logging of file-system-driver debug messages in the syslog for the specified virtual AIX client.	root

Command	Description	Role
"stfsdriver" on page 273	(AIX only) Loads the file-system driver as a kernel extension. <b>Note:</b> This command may be used by administrative users. Only specific options in this command are intended for use only by trained service technicians.	root
"stfsstat" on page 309	(AIX only) Display statistics about the client.	root

## disableConsoleTrace

Disables tracing and logging for the SAN File System console.

►►—disableConsoleTrace—◄◄

### Prerequisites

This task must be performed only by trained service technicians.

You must have root privileges to use the command.

The Web server must be running.

### Description

**Note:** This command is run from the shell prompt. It is not run inside of sfscli.

This command is case sensitive and must be entered as shown.

Tracing is automatically disabled after you restart the Web server.

### Example

**Disable tracing**The following example disables tracing and logging for the SAN File System console:

```
#disableConsoleTrace
Disabling SAN File System console tracing...
```

## enableConsoleTrace

Enables tracing and logging for the SAN File System console.

►►—enableConsoleTrace—◄◄

### Prerequisites

This task must be performed only by trained service technicians.

You must have root privileges to use the command.

The Web server must be running.

## Description

**Note:** This command is run from the shell prompt. It is not run inside of sfscli.

This command is case sensitive and must be entered as shown.

Tracing is automatically disabled after you restart the Web server.

This command save log and trace records to the /opt/was/logs/serverx/trace.log file.

Enabling tracing will impact the performance of SAN File System. Use this command only when necessary.

## Example

**Enable tracing** The following example enables tracing and logging for the SAN File System console:

```
enableConsoleTrace
Enabling SAN File System console tracing...
```

## mktruststore

Obtains a public certificate from LDAP to and creates a truststore that is shared by the administrative infrastructure to certify secure connections. This command replaces any existing truststore on the local engine.



## Parameters

*path*

Specifies the full directory path to import an existing LDAP certificate into the new truststore file being created. The certificate cannot be added to an existing truststore because a new truststore will be created. If not specified, an LDAP certificate will not be added to the new truststore.

**Note:** An LDAP certificate is required to obtain secure communication with the LDAP server.

- Specifies that you want to read the path from stdin (for example, - << /work/path.txt).

## Prerequisites

The cimom.properties file must be set up prior to using this command.

The tank.properties file must exist. It is used by the **mktruststore** command to determine the language being used.

This task must be performed only by trained service technicians.

## Description

**Note:** This command is run from the shell prompt from the `/usr/tank/admin/bin` directory. It is not run inside of `sfscli`.

The truststore file resides in the `/usr/tank/admin/` directory.

You would use the **mktruststore** command is used in these circumstances:

- During installation, the **mktruststore** command is run automatically by the **setupTank** script on the first engine. Because the truststore file must be exactly the same on every engine in the cluster, you must copy the truststore file from one engine to each remaining engine in the cluster before running **setupTank** on those engines.
- When replacing an expired truststore file. The truststore file is valid for one year.
- When you need to change the truststore (for example, if security is breached). You may change the truststore at any time; however, it must be the same on every engine in the cluster. The Administrative agent must be restarted any time the truststore is changed by issuing the **stopcimom** and **startcimom** commands. The user interfaces (SAN File System console and Administrative command-line interface) connection will be broken when the Administrative agent is stopped.

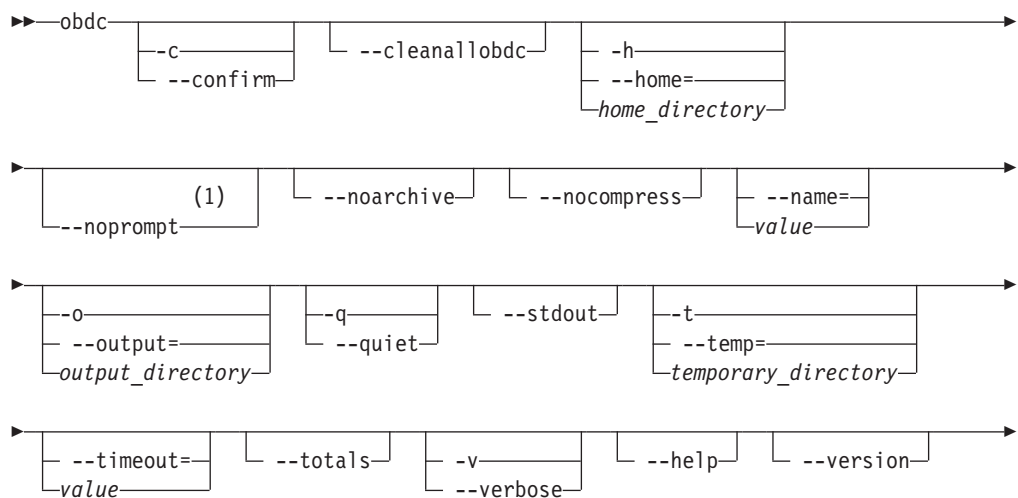
## Example

**Create truststore** The following example creates a truststore and imports the LDAP certificate `ldap.cert` from the local directory:

```
mktruststore ldap.cert
Creating truststore file.
The truststore was created successfully
Certificate was added to keystore.
```

## obdc

Collects information for the first-failure data capture of SAN File System problems, from either a client machine or the storage engine.





#### Notes:

- 1 When specifying the `--noprompt` option, the responses to all questions are automatically yes. Be sure that this will produce the expected results.

#### Parameters

##### `-c | --confirm`

Requests that the user be required to provide a confirmation for each piece of data that is collected.

##### `--cleanallobdc`

Removes all old one-button data collection (OBDC) archives and core files from the system. The user is asked to confirm each deletion.

##### `-h | --home= home_directory`

Specifies the root of the installation directory for the SAN File System component installed on the local machine. The default is `/user/tank`. The user can specify the home directory as a fully qualified path name only.

##### `--noprompt`

Specifies that the command will not prompt the user for the answers to questions. Instead, it is to assume that the answers to all questions are affirmative.

**Attention:** This option applies to all other specified options, including the `--cleanallobdc` option, so be sure that you consider the results.

##### `--noarchive`

Specifies that the command is to generate the data in a directory tree below the target directory, instead of archiving the data into a zip file. The name of the target directory is **OBDC-*mmddyy-time-odbc\_PID***, where *odbc\_PID* is the process ID for the OBDC command instance. For example, **OBDC-012204-1312-28494**.

##### `--nocompress`

Specifies that the command not compress the archive file. If you specify `--nocompress`, the name of the archive file is **OBDC-*mmddyy-time-odbc\_PID*.tar**, where for example, **OBDC-012204-1312-28494.tar** and if you do not specify `--nocompress`, the name of the archive file is **OBDC-*mmddyy-time-odbc\_PID*.tar.gz**, where *odbc\_PID* is the process ID for the OBDC command instance. For example, **OBDC-012204-1312-28494.tar.gz**.

##### `--name= value`

Assigns a specific name to the resulting tar file.

- o | --output= *output\_directory***  
Specifies the directory in which to assemble the collected data. The default is /user/tank/OBDC.
- q | --quiet**  
This mode does not display any output.
- stdout**  
Specifies that only information about how the command ran be sent to standard output.
- t | --temp= *temporary\_directory***  
Specifies an alternate directory in which to store temporary files. The default is /tmp.
- timeout *value***  
Specifies the amount of time, in seconds, before stopping an attempt to collect a single piece of data. The default is 30 seconds.
- totals**  
Displays the size and number of files acquired.
- v | --verbose**  
Enables verbose mode.
- help**  
Displays a detailed description of this command, including syntax, parameter descriptions, and examples. If you specify a help option, all other command options are ignored.
- version**  
Displays information about the version of the OBDC program.
- basic, config, crash, full, hardware, images, info, internal, logs, os, perf**  
Specifies the type and amount of data to be collected. You can specify more than one value, separated by a space between each type.
  - basic** Collects only a small amount of fundamental system data, which is useful for addressing simple usage and configuration questions. This information includes configuration files, the internal state of the metadata server or client, operating system statistics, hardware specifications and log files.  
  
This is the default collection option if no command-line options are specified.
  - config** Collects the configuration files for a metadata server or client.
  - crash** Collects crash dump files, such as operating system dumps, if they exist, or core files from the metadata server.
  - Tip:** Using the **crash** value can cause the **obdc** command to collect a very large amount of data. For example, a client machine with 2 GB of memory produces a core file of the same size. (The binary file for the operating system can also be somewhat large). Therefore, you must take steps to ensure the availability of sufficient disk space in the output directory before invoking the **obdc** command with this value.
  - full** Collects all of the data necessary to handle the majority of problems. It includes as the information collected with the basic options plus crash dump files and performance information.



**hardware**

Collects hardware specifications.

**images**

Collects installation files.

**info**

Collects information about the metadata server.

**internal**

Collects different types of runtime state information for either the client or the metadata server.

**logs**

Collects log files from the metadata server or client.

**os**

Collects operating system statistics.

**perf**

Gathers statistics and metrics about the performance of a metadata server or client, which is useful in cases where system response has degraded.

**Prerequisites**

This task must be performed only by trained service technicians.

**Description**

The **obdc** command gathers data for diagnosing errors or failures associated with metadata servers and clients. This command is intended primarily for first-failure data-capture capabilities useful for investigating problems upon their initial occurrence, without requiring problem recreation or subsequent tracing.

This command must be invoked natively on the metadata server or client machine from which you want to collect information. For a metadata server, this command can be invoked using the SAN File System console or the administrative command-line interface (CLI). The CLI command is named **collectdiag**. For a metadata server or client, the command can be invoked from a command shell (on UNIX) or a command prompt (on Windows). The SAN File System console provides true one-button running of the command against a specified metadata server, but it always issues the command with the default parameters for all parameters. Invoke the command using one of the alternate methods if you need to specify non-default parameters.

**Privileges**

The **obdc** command relies upon a number of SAN File System and operating system commands that require certain administrative privileges. To run the command, you must be logged in as root on a UNIX platform or Administrator on a Windows platform.

**Collected data**

The **obdc** command collects data for metadata servers, administrative servers, and clients of a SAN File System configuration, regardless of platform. The command relies both on SAN File System administrative commands and on operating-system commands to obtain much of the data it collects. It creates a GZIP-compressed tar file (by default) in the time-stamped output directory to record the results of the administrative and operating system commands that it invokes. In the event that the process from which data is requested is unavailable (for example, if the metadata server is unavailable or unresponsive), the command records a suitable

message and continues without collecting data from the unavailable process. This command also assembles various configuration, log, trace, and core files in the output directory.

The **obdc** command collects information related to both hardware and software. The specific information gathered depends on the platform, the SAN File System configuration of the machine, and the parameters specified. In general, this command collects more data for metadata servers than for clients, due to the greater function and complexity of the metadata server configuration.

For hardware, this command collects information about:

- Processors
- Memory
- Network adapters (LAN and SAN)
- Storage devices (local and SAN)

For software, this command collects information about:

- Operating system
- Network configuration (LAN and SAN)
- Storage configuration (local and SAN)
- SAN File System configuration

Within these general categories, the command gathers essential component information (for example, name, version, and configuration), current status, logged messages, trace and diagnostic output, core files, and performance statistics and metrics, among other things. Not all types of information pertain to all hardware and software components about which information is collected.

### Example

**Collecting basic information plus installation files** The following example collects the basic data and installation files. The collected data is archived and compressed into a single output file for delivery to remote support personnel.

```
/usr/tank/client/bin/obdc basic images
```

## startCimom

Starts the administrative agent.

▶▶—startCimom—◀◀

### Prerequisites

You must have root privileges to use this command.

This task must be performed only by trained service technicians.

### Description

**Note:** This command is run from the shell prompt. It is not run inside of sfscli.

This utility assumes that the required files are located in the `/usr/tank/admin` directory.

## Example

**Start the administrative agent** The following example starts the administrative agent:

```
/usr/tank/admin/bin/startCimom
```

## startConsole

Starts the SAN File System console Web server.

▶▶—startConsole—◀◀

### Prerequisites

You must have root privileges to use the command.

### Description

**Note:** This command is run from the shell prompt. It is not run inside of sfscli.

This command is case sensitive and must be entered as shown.

The Web server must be running for you to open the SAN File System console. The Web server starts automatically when the storage engine boots up. This command allows you to manually restart the Web server if it is stopped.

## Example

**Starts the Web server** The following example starts the SAN File System console Web server.:

```
startConsole
```

```
Starting the SAN File System console...
```

```
The SAN File System console is operational at https://189.24.15.162:7979/tank
```

## stfsdebug

Enables or disables the logging of file-system-driver debug messages in the syslog for the specified virtual client for AIX. Note that the equivalent command for the Solaris operating system is the **sanfs\_ctl trace** command.

▶▶—stfsdebug—  
┌ on ───┐ ┌ kmname kernel\_ext\_name ───┐ ┌ class debug\_class\_name ───┐  
└ off ─┘ └──────────────────────────┘ └──────────────────────────┘

### Parameters

**on** Enables the logging debug messages in the syslog.

**off**

Disables the logging of debug messages in the syslog.

**-kmname** *kernel\_ext\_name*

Identifies kernel-extension name of the file-system driver associated with the virtual client.

The file-system driver is loaded as a kernel extension. To identify the instance of the file-system drive, you identify the kernel extension. Each kernel

extension has a name, but this name is not unique. This name is usually the file name of the object file from which you loaded the kernel extension (for example, /usr/tank/client/bin/stfs).

**-class** *debug\_class\_name*

Identifies the class name for debug messages. If not specified, this command applies all classes. You can specify any of the following values:

**ATTR** File attribute information.

**CONFIG**

Driver initialization, termination, and control.

**DISK** Disk configuration at client-creation time

**IO** I/O operations.

**MOUNT**

FSIs and clients creation and destruction.

**PAGER**

Pager and items that give work to the pager.

**RDWR**

Reading from and writing to file cache.

**RNGLOCK**

Range lock information.

**VFSOP**

vfs operations, excl get root vnode

**VNODE**

Management of vnodes, gnodes, and stfsnodes.

**VNODEOP**

vnode operations, excl lookup, hold and release vnode.

## Prerequisites

This task must be performed only by trained service technicians.

## Example

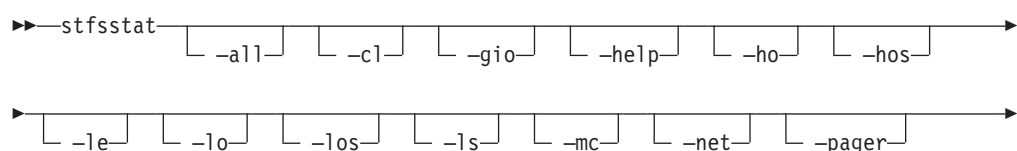
**Enable debugging on a client for AIX** The following example enables the logging of file-system-driver debug messages in the syslog for the specified virtual client:

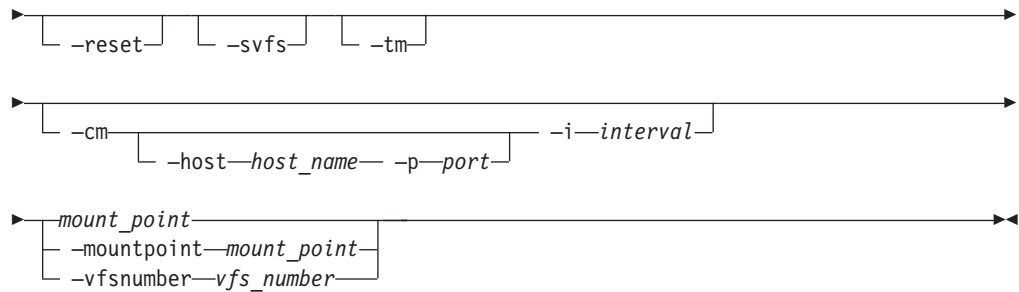
```
stfsdebug on -kmname /lib/stfs
```

## stfsstat

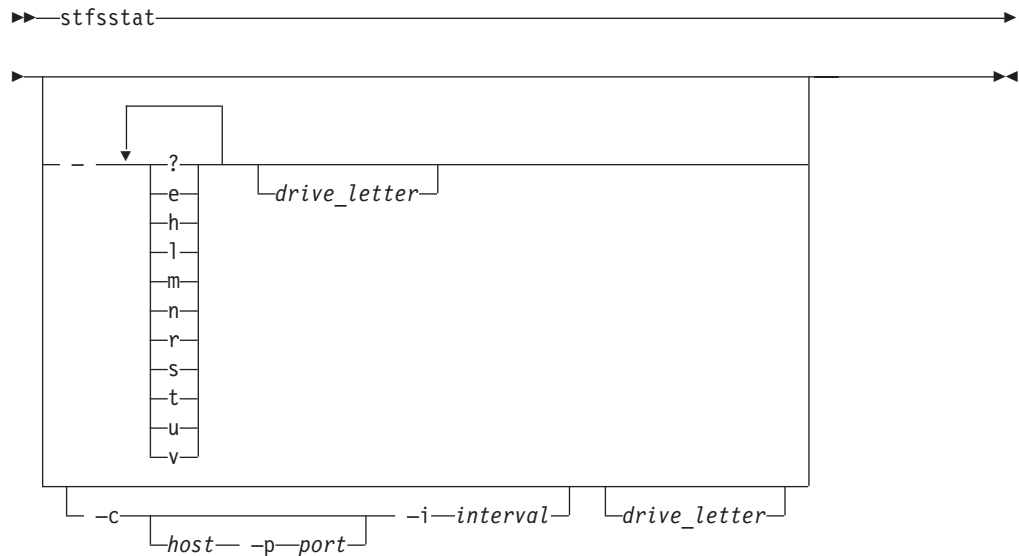
Displays statistics for the local client. Note that the **sanfs\_ctl stats** command is the equivalent command on the Solaris operating system.

### Clients for AIX





## Clients for Windows



## Parameters

### Clients for AIX

#### -all

Displays all client for AIX statistics, except per-object statistics. This parameter is the same as specifying `-tm`, `-mc`, `-ls`, `-le`, `-svfs`, `-gio`, `-pager`, and `-cl` parameters.

#### -cl

Displays cleaner statistics.

#### -cm [-host *host\_name* -p *port*] -i *interval*

Continuously monitors the specified statistics. The statistics are sent as a string to the specified *host\_name* on the specified UDP *port* at every specified *interval* (in seconds). The minimum interval is 20 seconds. If no host or port is specified, the statistics are sent to stdout.

#### -gio

Displays I/O statistics per global disk (GDISK), including : blocks read and written, and the average number of bytes read and written.

#### -help

Displays a detailed description for this command.

- ho**  
Lists the states of all objects in the mcObject hash table, including object ID, flags, and attributes
  - hos**  
Displays a short list of object hash table.
  - le**  
Lists mutex and latch state of all objects in the mcObject hash table, including current holder, hold state, and waiters.
  - lo**  
Lists the states of all objects in the mcObject LRU list, including object ID, flags, and attributes.
  - los**  
Displays a short list of object LRU.
  - ls**  
Lists mutex and latch statistics of metadata cache objects such as mcInstance and mcObject. Statistics include the number of attempts, wait time, and hold time.
  - mc**  
Displays metadata-cache statistics.
  - net**  
Displays network statistics per metadata server with which this client has a lease.
  - pager**  
Displays pager statistics per client, including information about the page-in and page-out activity.
  - reset**  
Resets all statistics to zero. Some statistics, such as object and latch state, cannot be reset because no counters are incremented, but data structures are shown.
  - svfs**  
Displays file-system statistics. These metrics are mount-specific and distinguish between the different mount points. Statistics include the number of file, inode, address space, and super operations performed.
  - tm**  
Displays Transaction Manager (TM) statistics per client related to TM data structures, including the number of messages sent and received (per message type), the maximum lengths and average lengths of the transaction queue, and the number of transactions, messages, and leases lost.
- mount\_point* | **-mountpoint** *mount\_point*  
The mount point where you want to mount the file-system image (for example, the directory above which you want the global namespace image's directory tree to appear).
- vfsnumber** *vfs\_number*  
Identifies the virtual-file-system (VFS) number associated with the global namespace image for which you want to display statistics. In AIX, every global namespace image has a unique VFS number.
- The **stfsmount** command displays this number when it creates the global namespace image.

## Clients for Windows

- ?** Displays a detailed description for this command.
- c** Continuously monitors the specified statistics.
- e** Displays the latch state.
- h** Displays the object hashlist statistics.
- i** Displays the interval for sending statistics. This option must be used with the **-c** parameter.
- l** Lists mutex and latch statistics of metadata cache objects.
- m**  
Displays metadata-cache statistics.
- n** Displays the engine statistics.
- p** Displays the port for sending statistics. This option must be used with the **-c** parameter.
- r** Resets the CSM statistics
- s** Displays the object LRU statistics.
- t** Displays Transaction Manager (TM) statistics. This is the default option.
- u** Displays statistics in the short-output format.
- v** Displays file-system statistics

### *drive\_letter*

The drive letter mapped to SAN File System. The default letter is T.

## Prerequisites

This task must be performed only by trained service technicians.

## Description

On Windows machines running the client, the statistics parameters (**?**, **e**, **h**, **l**, **m**, **n**, **r**, **s**, **t**, **u**, and **v**) are appended together. For example, to choose to display the latch state and statistics, you would specify **-el**.

## Example

**Displays statistics from a client for AIX** The following example displays transaction-manager statistics and metadata-cache statistics for the local client for AIX that is mounted on `/mnt/sanfs`:

```
stfsstat -tm -mc /mnt/sanfs
```

**Displays statistics from a client for Windows** The following example displays statistics for the object-hash list and latch state the local client for Windows mapped to drive S:

```
stfsstat -he s
```

## stopCimom

Stops the administrative agent.

### Prerequisites

You must have root privileges to use this command.

This task must be performed only by trained service technicians.

### Description

**Note:** This command is run from the shell prompt. It is not run inside of sfscli.

This utility assumes that the required files are located in the /usr/tank/admin directory.

### Example

**Stop the administrative agent** The following example stops the administrative agent:

```
/usr/tank/admin/bin/stopCimom
```

## stopConsole

Stops the SAN File System console Web server.

### Prerequisites

You must have root privileges to use the command.

### Description

**Note:** This command is run from the shell prompt. It is not run inside of sfscli.

This command is case sensitive and must be entered as shown.

You would stop the SAN File System console Web server when you are upgrading software.

### Example

**Stops the Web server** The following example stops the SAN File System console Web server:

```
stopConsole
Stopping the SAN File System console...
```

## tank extractbootrecord

Extracts the product (disk) label information contained in the master volume and creates a local copy of the Tank.Bootstrap file.

```
└─device—device_path─┘
```



## Parameters

**-device** *device\_path*

Specifies the device path for a valid master volume from to read the product label.

## Prerequisites

This task must be performed only by trained service technicians.

The cluster must be offline.

## Description

This command is located in the `/usr/tank/server/bin` directory.

This command is used in disaster recovery situations or when attaching new hardware to an existing set of SAN devices.

Each metadata server in the cluster stores a local copy of the product label that is on the master volume. The local copy is stored in the file `/etc/tank/server/Tank.Bootstrap`, in binary format. SAN File System uses this information to locate the master volume on the SAN at boot time. After it is located, the cluster definition can be read. You would use this command when the local copy of the `Tank.Bootstrap` file is lost on all engines to extract the product label from the system master volume and recreate the local `Tank.Bootstrap` file.

This command is used to recreate the local `Tank.Bootstrap` file on the master metadata server in a single-engine cluster. The `Tank.Bootstrap` file is created on each remaining metadata server as they are added to the cluster (using the `addserver` command). It is not necessary or recommended to run this command on every metadata server or to copy this file from one metadata server to another.

**Note:** This command is run from the shell prompt. It is not run inside of `sfscli`.

This command allows a single-engine cluster to be brought up on new hardware in the case of disaster recovery or when you attach new hardware to an existing cluster. Recovering the `Tank.Bootstrap` file is required only when copies on all engine hosting metadata servers in the cluster are lost.

## Example

**Extract a boot record** The following example extracts a local `Tank.Bootstrap` file from device `/dev/rsdc`. This device is a valid master volume of a SAN File System cluster instance.

```
#/usr/tank/server/tank extractbootrecord -device /dev/rsdc
Label information from master disk /dev/rsdc was extracted and stored in
Tank.Bootstrap.
```

## tank lscluster

Displays the cluster definition from a system master volume when the metadata servers are not running.

►►—tank—lscluster—◄◄

## Prerequisites

This task must be performed only by trained service technicians.

The cluster must be offline.

## Description

**Note:** This command is run from the shell prompt. It is not run inside of sfscli.

This command is used in disaster recovery situations, while troubleshooting the system, or to inspect the cluster definition.

This command is located in the `/usr/tank/server/bin` directory.

This command displays the following information:

- Master state of the state machine in the cluster services component.
- Subordinate state of the state machine in the cluster services component.
- Transport protocol used for group services. The value defaults to UDP and cannot be changed. Note that the group services includes cluster services.
- Transition state, which indicates whether a cluster transition is in progress at the time this command is issued.
- Started state, which indicates whether the group services subsystem was started.
- Joining state, which indicates whether the metadata server attempted to join the cluster.
- Forming state, which indicates whether the metadata server is attempting to reform the cluster as the master metadata server.
- Dynamic group ID, which identifies the subset of the cluster that is alive and well.
- Time when the cluster was last reformed (committed).
- Size of the current dynamic group. This is affected when the cluster or a metadata server starts, stops, crashes, or aborts.
- Size of the entire static cluster. This is affected by the **addserver** and **dropserver** commands.
- Cluster identifier, which is initialized at installation time.
- Cluster name (sanfs).
- Time of the installation.
- Timeout value used by group services.
- IP address of the network used by the cluster.
- Netmask used by the cluster.
- Current active version of the cluster boot record. Two copies are maintained. This value will be A or B.
- Identifier of the engine hosting the metadata server where this command was issued.
- A list of all engines in the cluster and attributes for each. The attributes include:
  - Engine identifier.
  - IP address.
  - Group services port number (gs)
  - SAN File System protocol port number (stp).
  - Heartbeat port number (hb).
  - Administrative port number (adm)

- Administrative agent port number (agent)
- Metadata server/engine name.
- Last committed software version.

### Example

**Lists the static cluster definition** The following example lists the static cluster definition before running the **tank resetcluster** command:

```
tank lscluster
Group information:
mast_state: Microkernel
sub_state: Invalid
protocol: udp
in transition: no
b_started: 0
b_joining: 0
b_forming: 0
group_id: 9
group commit time: Jun 25, 2003 9:47:53 PM
group size: 4
cluster size: 4
cluster id: 1234
cluster name: sanfs
install time: Jun 25, 2003 4:23:45 PM
installation id: 835407414733488113
gs_timeout: 1000 (millisecs)
ip network: 0.0.0.0 (interfaces:)
netmask: 0.0.0.0
active set: B
this node id: 0
```

	id	ip addr	gs	stp	hb	gdm	agent
NODE: 0		192.168.10.88	11003	11001	11004	11002	5989
NODE: 1		192.168.11.88	11003	11001	11004	11002	5989

stnode name	SW Version
GR ST0	1.0.1
GR ST1	1.0.1

## tank lsversion

Displays the version control information from a system master disk.

```
▶▶—tank—lsversion—◀◀
```

### Prerequisites

This task must be performed only by trained service technicians.

This command must be used only under the direction of your IBM support representative.

### Description

You would use this command only during troubleshooting or disaster recovery situations.

## Example

**Displays version control information** The following example version control information from a system master disk:

```
/usr/tank/server/bin/tank lsersion
```

## tank lsdisklabel

Displays the SAN File System product (disk) label for a specified device.

►►—tank—lsdisklabel— -device—*device\_path*—◄◄

### Parameters

**-device** *device\_path*

Specifies the device path of a valid master volume from which to read a product label. The specified device path can include the path of the raw device as an argument, that is, *rvpathn* where *n* is the vpath letter. If the device path is not specified, the command might return inaccurate information.

### Prerequisites

This task must be performed only by trained service technicians.

The cluster must be offline.

### Description

This command is run from the shell prompt. It is not run inside of a sfscli session.

This command is used in disaster recovery situations, when you attach new hardware to an existing cluster, or while troubleshooting the system.

This command is located in the `/usr/tank/server/bin` directory.

If you display the product label from the master volume, this command displays this information:

- Label identifier 1. This is always **SDISK** for all SAN File System devices. This should match the label identifier 2. It is used as an integrity check to detect valid or corrupted product labels.
- Label number.
- Disk type. Valid values are:
  - M** Master volume label
  - S** System volume label
  - U** User volume
- Global disk identifier. This is the ID of volume that is used and understood by the metadata servers and clients.
- Label date. This is the date when the volume was added.
- Owner identifier. This is the ID of the cluster that owns the volume.
- Installation identifier. This is the unique installation ID for the shared storage that is initialized during installation and changes only upon reinstallation.
- Disk epoch identifier.
- Product identifier.

- Label identifier 2. This is always **SDISK** for all SAN File System devices. This should match the label identifier 1. It is used as an integrity check to detect valid or corrupted product labels.

If you display the product label from a local Tank.Bootstrap file, this command displays this information:

- Disk type.
- Global disk identifier.
- Owner identifier.
- Installation identifier.
- Disk epoch identifier.

**Tip:** The owner ID, installation ID, and disk epoch ID for a given volume is the same as the owner ID, installation ID, and disk epoch ID contained in the Tank.Bootstrap file on all engines in the cluster that own that volume. An engine can be zoned in such a way that it sees devices from another cluster, but each cluster uses the product label information in the Tank.Bootstrap file to know which devices it owns.

### Example

**Displays the product label for the system master volume** The following example displays the product label for the system master volume contained on device/dev/rvpatha (a is the vpath letter):

```
/usr/tank/server/bin/tank lsdisklabel -device /dev/rvpatha
```

```

Product Label on Device /dev/rvpatha:

```

```
Label ID 1 : SDISK
Label Number : 0001
Disk Type : M
Global Disk ID : 73EFA22D6A2355F1
Label Date : Jun 25, 2003 4:35:24 AM
Owner ID : 00000000000004D2
Installation ID : 73EFA22D6A2355F1
Disk Epoch : 0000000000000000
Product ID : STORAGETANK
Label ID 2 : SDISK

```

**Displays the information in Tank.Bootstrap** The following example displays the product label information found in the local Tank.Bootstrap file:

```
/usr/tank/server/bin/tank lsdisklabel
```

```

Tank.Bootstrap information

```

```
Disk Type : U
Global Disk ID : 73F02ABDBE27B00B
Owner ID : 00000000000004D2
Installation ID : 73F02ABDBE27B00B
Disk Epoch : 0000000000000000

```

## tank resetcluster

Erases the static cluster definition contained on the system master volume, without reinitializing the metadata, which in affect drops all metadata servers from the cluster at one time.

►►—tank—resetcluster—◄◄

### Prerequisites

This task must be performed only by trained service technicians.

The cluster must be offline.

### Description

**Note:** This command is run from the shell prompt. It is not run inside of sfscli.

**Attention:** This command will wipe out the static cluster definition contained on the master volume. It is equivalent to issuing the **dropserver** command on all engines.

This command is used in extreme disaster recovery situations or when you attach new hardware to an existing cluster.

This command is located in the /usr/tank/server/bin directory.

This command must be run from an engine that has a Tank.Bootstrap file. If the engine does not have a Tank.Bootstrap file, this file can be recovered from the master volume using the **tank extractbootrecord** command. If the master volume is unknown, you can issue the **tank lsdisklabel** command to inspect the SAN File System product label and to locate the master volume.

After you run this command, the first metadata server that is started with a valid Tank.Bootstrap file becomes the new master metadata server. At this point, the cluster is a single-engine cluster, and all additional metadata servers must be added using the **addserver** command. In addition, it might be necessary to move filesets to these new engines (using the **setfilesetserver** command) if the added engines have different names than the original cluster's engine names.

### Example

**Reset the cluster** The following example erases the static cluster definition on the master volume:

```
#/usr/tank/server/bin/tank resetcluster
Static cluster definition has been reset. cluster size = 0
```

## tank resetversion

Resets the version-control information on a system master disk.

### Syntax

►►—tank—resetversion—◄◄

### Prerequisites

This task must be performed only by trained service technicians.

The cluster must be offline.

### Description

**Attention:** This command should be used only by trained service technicians and only in the event of extreme disaster recovery or hardware refresh situations. Invoking this command at runtime can lead to unpredictable and destructive results.

### Example

**Resets the version-control information** The following example resets the version-control information on a system master disk:

```
/usr/tank/server/bin/tank resetversion
```

---

## Appendix C. Environment variables

This topic provides a list of environment variables in SAN File System.

### Administrative CLI variables

The following environment variables modify the behavior of the administrative CLI:

**ADMINDIR** = *directory*

Specifies the directory where the administrative binaries and the truststore are located. The default directory is /usr/tank/admin.

**SFS\_CLI\_PASSWDFILE** = *directory/file*

Specifies the full-qualified name of the password file that stores the user ID and password information.

**SFS\_CLI\_VERBOSE** = *flag*

Turns verbose mode on or off. You can specify a value of ON or OFF.

**TANKDIR**=*directory*

Specifies the directory in which to save diagnostic results.





---

## Appendix D. File placement policy syntax

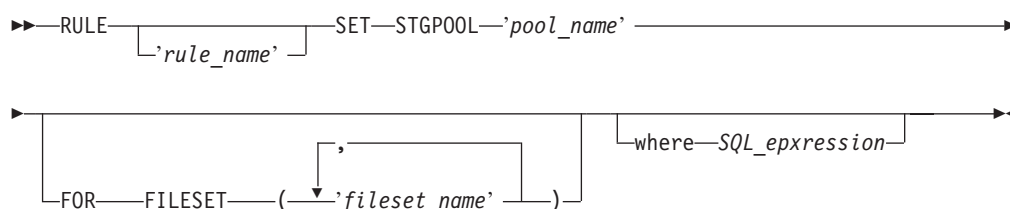
This topic describes the syntax conventions for file-placement rules.

You can create a file containing policy rules for placing newly created files. You can then use this rule file when creating a policy using the **mkpolicy** command from the administrative CLI. You can also edit the policy rules that you create using the SAN File System console.

### Note:

1. Every policy file must start with VERSION 1.
2. A policy is not required to contain any rules, in which case it would be equivalent to the default policy.
3. The maximum size of a policy is 32 KB.

You can also add comments to the policy. All comments must start with `/*` and end with `*/` (for example, `/* comment */`).



### Parameters

#### RULE

Initiates the rule statement.

*'rule\_name'*

Identifies the rule. This parameter is optional.

#### SETSTGPPOOL *'pool\_name'*

Identifies the pool in which you want to place all files that match the rule criteria (fileset and SQL expression).

#### FOR FILESET (*'fileset\_name'*)

Identifies one or more filesets in which the file is created to determine where the file is to be placed. In the case of nested filesets, the rules apply if the file is created in the innermost fileset.

#### where *SQL\_expression*

Compares the file attributes specified in the rule with the attributes of the file being created to determine where the file is to be placed. The *SQL\_expression* can be any combination of standard SQL-syntax expressions, including comparison predicates, between predicates, in predicates, like predicates, mathematical value expressions, and boolean, string and numeric literals.

**Note:** Case expressions and compared-when clauses are not allowed.

SAN File System supports built-in functions, which can be used in comparison predicates, between predicates, in predicates, and like predicates. These

functions are organized in three categories: date and time manipulation, numeric calculations, and string manipulation.

### Attributes

You can use any of these attributes in the expression:

#### NAME

Name of the file. You can use a % wildcard in this name to represent zero or more characters and use the \_ wildcard to represent a single character.

#### CREATION\_DATE

Date and time that the file was created.

#### GROUP\_ID

Numeric group ID. This attribute is valid only for AIX clients.

#### USER\_ID

Numeric user ID. This attribute is valid only for AIX clients.

### String functions

You can use these string-manipulation functions on file names and literals.

**Note:** You must enclose strings in single-quotation marks. You may include a single-quotation mark in a string by using two single-quotation marks (for example, 'a''b' represents the string a'b).

#### CHAR(*x*)

Converts an integer *x* to a string.

#### CHARACTER\_LENGTH(*x*)

Determines the number of characters in string *x*.

#### CHAR\_LENGTH(*x*)

Determines the number of characters in string *x*.

#### CONCAT(*x*,*y*)

Concatenates strings *x* and *y*.

#### HEX(*x*)

Converts an integer *x* in hexadecimal format.

#### LCASE(*x*)

Converts string *x* to lowercase.

#### LEFT(*x*,*y*,*z*)

Left justifies string *x* in a field of *y* characters, optionally padding with character *z*.

#### LENGTH(*x*)

Determines the length of the data type of string *x*.

#### LOWER(*x*)

Converts string *x* to lowercase.

#### LTRIM(*x*)

Removes leading blank characters from string *x*.

#### POSITION(*x* IN *y*)

Determines the position of string *x* in string *y*.

#### POSSTR(*x*,*y*)

Determines the position of string *y* in string *x*.

**RIGHT(*x,y,z*)**

Right justifies string *x* in a field of *y* characters, optionally padding with character *z*.

**RTRIM(*x*)**

Removes the trailing blank characters from string *x*.

**SUBSTR(*x* FROM *y* FOR *z*)**

Extracts a portion of string *x*, starting at position *y*, optionally for *z* characters (otherwise to the end of the string).

**SUBSTRING(*x* FROM *y* FOR *z*)**

Extracts a portion of string *x*, starting at position *y*, optionally for *z* characters (otherwise to the end of the string).

**TRIM(*x*)**

Trims blank characters from the beginning and end of string *x*.

**TRIM(*x* FROM *y*)**

Trims blank characters that are *x* (LEADING, TRAILING, or BOTH) from string *z*.

**TRIM(*x y* FROM *z*)**

Trims character *y* that is *x* (LEADING, TRAILING, or BOTH) from string *z*.

**UCASE(*x*)**

Converts the string *x* to uppercase.

**UPPER(*x*)**

Converts the string *x* to uppercase.

**Numerical functions**

You can use these numeric-calculation functions to place files based on either numeric parts of the file name, numeric parts of the current date, and AIX-client user IDs or group IDs. These can be used in combination with comparison predicates and mathematical infix operators (such as addition, subtraction, multiplication, division, modulo division, and exponentiation).

**INT(*x*)**

Converts number *x* to a whole number, rounding up fractions of .5 or greater.

**INTEGER(*x*)**

Converts number *x* to a whole number, rounding up fractions of .5 or greater.

**MOD(*x,y*)**

Determines  $x \% y$ .

**Date and time functions**

You can use these date-manipulation and time-manipulation functions to place files based on when the files are created at the client and the local time of the subordinate metadata server serving the directory within which the file is being created.

**CURRENT DATE**

Determines the current date on the subordinate metadata server.

**CURRENT\_DATE**

Determines the current date on the subordinate metadata server

**CURRENT TIME**

Determines the current time on the subordinate metadata server.

**CURRENT\_TIME**

Determines the current time on the subordinate metadata server.

**CURRENT TIMESTAMP**

Determines the current date and time on the subordinate metadata server.

**CURRENT\_TIMESTAMP**

Determines the current date and time on the subordinate metadata server.

**DATE(*x*)**

Creates a date out of *x*.

**DAY(*x*)**

Creates a day of the month out of *x*.

**DAYOFWEEK(*x*)**

Creates the day of the week out of date *x*, where *x* is a number from 1 to 7 (Sunday=1).

**DAYOFYEAR(*x*)**

Creates the day of the year out of date *x*, where *x* is a number from 1 to 366.

**DAYS(*x*)**

Determines the number of days since 0000-00-00.

**DAYSINMONTH(*x*)**

Determines the number of days in the month from date *x*.

**DAYSINYEAR(*x*)**

Determines the day of the year from date *x*.

**HOUR(*x*)**

Determines the hour of the day (a value from 0 to 23) of time or timestamp *x*.

**MINUTE(*x*)**

Determines the minute from date *x*.

**MONTH(*x*)**

Determines the month of the year from date *x*.

**QUARTER(*x*)**

Determines the quarter of year from date *x*, where *x* is a number from 1 to 4 (for example, January, February, and March is quarter 1).

**SECOND(*x*)**

Returns the seconds portion of time *x*.

**TIME(*x*)**

Displays *x* in a time format.

**TIMESTAMP(*x*,*y*)**

Creates a timestamp (date and time) from a date *x* and optionally a time *y*.

**WEEK(*x*)**

Determines the week of the year from date *x*.

**YEAR(*x*)**

Determines the year from date *x*.

## Time and dates formats

Use any of the these formats when specifying times and dates.

**Note:** All date and time attributes in these rules are based in coordinated universal time (UTC).

### Timestamp

Use one of the following formats to specify a timestamp:

- *date time*
- *date*

There must be exactly one space between the date and time.

You can mix formats for the date and time. For example, you can specify ISO format for the date and international format for the time.

**Date** Use one of these formats to specify a date:

#### European

*DD.MM.YYYY*

**ISO** *YYYY-MM-DD*

**USA** *MM/DD/YYYY*

You may leave off leading zeros from *MM* (month) and *DD* (day). You can use a two-digit year, in which case 1900 is added if the year is greater than 50 and 2000 is added if the year is 50 or less.

**Note:** The `MONTHNAME()` and `DAYNAME()` functions produce English names with no internationalization.

**Time** Use one of these formats to specify a time:

#### International

*HH:MM[SS[.UUUUUU]]*

**USA** *HH[:MM[:SS]] [A|P|AM|PM]*

You may leave off leading zeros from any field except subseconds. The international format uses a 24-hour clock. The USA format uses a 12-hour clock followed by A, P, AM, or PM.

You can substitute commas or periods for colon delimiters in the international format.

## Examples

The following example shows a sample file.

VERSION 1

```
rule 'stgRule1' set stgpool 'pool1' for fileset ('cnt_A')
rule 'stgRule2' set stgpool 'pool2' where NAME like '%.doc'
rule 'stgRule3' set stgpool 'pool3' where DAYOFWEEK(CREATION_TIME) == 1
rule 'stgRule4' set stgpool 'pool4' where USER_ID <= 100
```



---

## Appendix E. Getting help, service, and information

If you need help, service, technical assistance, or just want more information about IBM products, you will find a wide variety of sources available from IBM to assist you.

IBM maintains pages on the World Wide Web where you can get information about IBM products and services and find the latest technical information.

Table 3 lists some of these pages.

*Table 3. IBM Web sites for help, services, and information*

<a href="http://www.ibm.com/">www.ibm.com/</a>	Main IBM home page
<a href="http://www.ibm.com/storage/">www.ibm.com/storage/</a>	IBM Storage home page
<a href="http://www.ibm.com/storage/support">www.ibm.com/storage/support</a>	IBM Support home page

Services available and telephone numbers listed are subject to change without notice.

### Software Maintenance

All distributed software licenses include Software Maintenance (software subscription and technical support) for a period of 12 months from the date of acquisition providing a streamlined way to acquire IBM software and assure technical support coverage for all licenses. Extending coverage for a total of three years from date of acquisition may be elected. While your Software Maintenance is in effect, IBM will provide you assistance for your 1) routine, short duration installation and usage (how-to) questions; and 2) code-related questions. IBM provides assistance via telephone and, if available, electronic access, only to your information systems (IS) technical support personnel during the normal business hours (published prime shift hours) of your IBM support center. (This assistance is not available to your end users.) IBM provides Severity 1 assistance 24 hours a day, every day of the year.

---

## Before you call for service

This topic provides information you need to know before you call for service.

Some problems can be solved without outside assistance, by using the online help, by looking in the online or printed documentation that comes with the SAN File System, or by consulting the support Web page noted in Table 3. Also, be sure to read the information in any README files and release notes that come with the SAN File System.

---

## Getting help online

Be sure to visit the support page for the SAN File System, complete with FAQs, parts information, technical hints and tips, technical publications, and downloadable files, if applicable. This page is at: [www.ibm.com/storage/support](http://www.ibm.com/storage/support).



---

## Getting help by telephone

With the original purchase of the SAN File System, you have access to extensive support coverage. During the product warranty period, you may call the IBM Support Center (1 800 426-7378 in the U.S.) for product assistance covered under the terms of the software maintenance contract that comes with SAN File System purchase.

Please have the following information ready when you call:

- SAN File System software identifier, which can be either the product name (SAN File System) or the Product Identification (PID) number
- Description of the problem
- Exact wording of any error messages
- Hardware and software configuration information

If possible, have access to your master console when you call.

In the U.S. and Canada, these services are available 24 hours a day, 7 days a week. In the U.K., these services are available Monday<sup>™</sup> through Friday, from 9:00 a.m. to 6:00 p.m. In all other countries, contact your IBM reseller or IBM marketing representative.<sup>1</sup>

---

1. Response time will vary depending on the number and complexity of incoming calls.

---

## Appendix F. Logs

This topic describes the logs that you can view from either the administrative command-line interface or the SAN File System console.

You can display the following logs from either the administrative command-line interface or the SAN File System console.

- Administrative log
- Audit log
- Cluster log
- Event log
- Security log

**Note:** The cluster log is a consolidated view of the server log on each engine in the cluster. The event log is a subset of the messages stored in the cluster log; it contains only those messages that have an message type of event.

---

### Administrative log

The administrative log contains messages generated by the administrative server.

If you display the administrative log from either the administrative command-line interface or the SAN File System console, all administrative logs on all engines in the cluster are consolidated into a single view.

#### Fields

Log entries contain the following fields:

##### Message ID

A unique identifier for the message.

##### Severity level

Indicates whether the entry is an informational, warning, error, or severe message.

##### Message type

Specifies whether the message is a normal log entry or one that was generated as a result of an event on the administrative server.

##### Administrative server ID

A unique identifier for the administrative server on which the command was issued.

##### Timestamp

A date followed by a local time.

##### Message

The log message.

The following example illustrates the consolidated view of the administrative log that is displayed through the administrative command-line interface:

```
CIMServer: Info Normal mds_engine_0 May 16, 2003 7:27:33 AM
 Namespace \root\cimv2 initialized
CMMOM0411I Info Normal mds_engine_0 May 16, 2003 7:27:33 AM
```

```

Authorization is not active
CMMOM0901I Info Normal mds_engine_0 May 16, 2003 7:27:33 AM
IndicationProcessor started
CMMOM0906I Info Normal mds_engine_0 May 16, 2003 7:27:33 AM
No pre-existing indication subscriptions
CMMOM0404I Info Normal mds_engine_0 May 16, 2003 7:27:33 AM
Security server starting on port 5989
CMMOM0402I Info Normal mds_engine_0 May 16, 2003 7:27:33 AM
Platform is Unix

```

---

## Audit log

This topic describes the contents of the audit log.

The administrative audit log contains all administrative actions issued to the SAN File System. It contains a record of every command issued by a SAN File System administrator, from either the administrative command-line interface or the SAN File System console, that changes the state of the metadata server in some way.

Each administrative server has its own audit log. If you display the audit log from either the administrative command-line interface or the SAN File System console, all audit logs on all engines in the cluster are consolidated into a single view.

### Fields

Log entries contain the following fields:

#### Timestamp

A date followed by a local time.

#### Severity level

Set to a value of Informational (console) if the command succeeded.  
Otherwise, it is set to a value of Error.

#### Message ID

A unique identifier for the message.

#### Message type

Set to Audit. This field is contained in the audit log, but it is not displayed in the consolidated view.

#### Metadata server ID

A unique identifier for the metadata server on which the command was issued.

#### Message

Contains the user name of the SAN File System administrator who issued the command followed by a functional replica of the log message itself.

The following example shows a message from the consolidated view of the audit log that is displayed through the administrative command-line interface:

```

2003-04-21 18:36:32 INFORMATIONAL HSTAD0083I A mds_engine_0
User Name: jozvold Command Name: MasterServiceAddServer
Parameters: SYSTEMCREATIONCLASSNAME=STC_Cluster
SYSTEMNAME=testnode CREATIONCLASSNAME=STC_MasterService
NAME=MasterService CLUSTERPORT=10001 IP=9.29.25.136
Command Succeeded.

```

---

## Event log

The event log contains all messages from the cluster message log that have a message type of event.

The event log contains all messages from the cluster message log that have a message type of event.

### Fields

Log entries contain the following fields:

#### Message ID

A unique identifier for the message.

#### Severity level

The severity indicates whether the entry is an informational, warning, or error message.

#### Metadata server ID

A unique identifier for the Metadata server on which the command was issued.

#### Timestamp

The timestamp consists of a date followed by a local time.

#### Message

The message is a textual explanation.

The following example illustrates the event log that is displayed through the Administrative command-line interface:

ID	Level	Server	Date and Time	Message
TANCM0393I	Info	svc-mds1	May 16, 2003 2:08:01 AM	ALERT: The server state has changed from Initializing(2) to Joining(5).
TANCM0393I	Info	svc-mds1	May 16, 2003 2:08:13 AM	ALERT: The server state has changed from Joining(5) to Online(10).
TANCM0389I	Info	svc-mds1	May 16, 2003 2:08:13 AM	ALERT: The cluster state has changed from Forming(6) to Online(10).
TANCM0393I	Info	svc-mds2	May 16, 2003 2:09:22 AM	ALERT: The server state has changed from NotAdded(4) to Joining(5)

---

## Security log

The security log displays the administrative user login activity for the administrative server.

The security log displays the administrative user login activity for the administrative server. If you display the security log from either the administrative command-line interface or the SAN File System console, all security logs on all engines in the cluster are consolidated into a single view.

### Fields

Log entries contain the following fields:

#### Message ID

A unique identifier for the message.

#### Severity level

Indicates whether the entry is an informational, warning, or error message.

**Administrative server ID**

A unique identifier for the administrative server on which the command was issued.

**Message**

The log message.

The following example illustrates the administrative log displayed through the administrative command-line interface:

```
CMMOM0302I Info mds_engine_0 May 19, 2003 9:21:17 AM
User respey on client {1} could not be authenticated
CMMOM0302I Info mds_engine_0 Jun 13, 2003 1:51:40 PM
User jkaminski on client {1} could not be authenticated
CMMOM0302I Info mds_engine_0 Jun 20, 2003 5:41:36 PM
User fstock on client {1} could not be authenticated
```

---

## Server log

The server message log contains operational information for the metadata server.

The server log contains operational information. Each metadata server in the SAN File System cluster has its own log. If you display the server log from either the administrative command-line interface or the SAN File System console, all server logs on all engines in the cluster are consolidated into a single view. The consolidated view of the server message log is called the cluster log.

**Fields**

Log entries contain the following fields:

**Timestamp**

A date followed by a local time.

**Severity level**

Indicates whether the entry is an informational, warning, error, or severe message.

**Message ID**

A unique identifier for the message.

**Message type**

Specifies whether the message is a normal log entry or one that was generated as a result of an event on the metadata server.

**Metadata server ID**

A unique identifier for the metadata server on which the command was issued.

**Message**

A textual explanation of the message.

The following example shows a message from the cluster log that is displayed through the administrative command-line interface:

```
2003-04-16 12:55:50 INFORMATIONAL HSTPG0009I N
msd_engine_0 Using IP 9.38.203.26 port 10192
for Group Services messages.
```

---

## Appendix G. SAN File System console panels

---

### Add Server to the Cluster panel

This panel allows you to add a metadata server to a cluster. The metadata server must be in "not added" state and should not appear in the Servers table.

#### Fields

##### IP Address

Specifies the IP address of the metadata server.

---

### Add Volumes to a Storage Pool - Add Volumes panel

This panel (the third step in the **Add Volumes to a Storage Pool** wizard) allows you to add available logical unit numbers (LUNs) as volumes to the selected storage pool. You can select from a list of available metadata LUNs if you are adding volumes to a system storage pool and available data LUNs if you are adding volumes to a user storage pool.

#### Attributes

The Available LUNs table displays the available LUNs from which you select, in order to write to the storage pool as volumes. It displays the size of the LUN in megabytes, as well as the following information:

##### LUN ID

The identifier for the LUN. When you click on a LUN ID in the table, you link to additional details about the LUN.

**Tip:** If you enter the **Add Volumes** wizard from the **Available LUNs** panel, your previously chosen LUN or LUNs are also preselected.

##### Vendor

The vendor of the storage device associated with the LUN.

##### Product

The product name of the storage device associated with the LUN.

##### OS Device Path

The path that points to the SAN Adapter driver.

---

### Add Volumes to a Storage Pool - Introduction panel

This panel introduces the **Add Volumes to a Storage Pool** wizard, which creates new volumes from available logical unit numbers (LUNs) and adds them to a SAN File System storage pool.

This wizard allows you to select the storage pool, access information about available LUNs, add the volume or volumes, and verify your settings before they are added.

**Note:** The **Next** button in the wizard is disabled if any failures occur during the add volume process.

---

## Add Volumes to a Storage Pool - Select Client panel

This panel (the second step in the **Add Volumes to a Storage Pool** wizard) allows you to specify settings in order to access information about logical unit numbers (LUNs) that are available to be added to a user storage pool as volumes.

### Fields

**Client** Selects a client that can see the available LUNs.

### Method of fetching the LUN information

Selects the method to use in order to fetch the LUN information from cache or rediscover the LUNs.

---

## Add Volumes to a Storage Pool - Select Storage Pool panel

This panel (the first step in the **Add Volumes to a Storage Pool** wizard) allows you to select a storage pool to which to add available LUNs as volumes.

### Attributes

In addition to its description, the selection table provides the following information about each storage pool:

**Name** Name of the storage pool. When you click on a storage pool name in the table, you link to additional details about the associated storage pool.

**Type** One of three storage pool types are possible:

#### User storage pools

Data storage pools in which user data gets stored in blocks.

#### User Default storage pool

The user storage pool that data gets assigned to, if not picked up by a policy rule.

#### System storage pool

The storage pool that stores the metadata (regarding filesets, policies, and so forth).

**Note:** It is important to ensure that this storage pool never runs out of space.

### Size (MB)

Total space available in the associated storage pool.

### Used (MB)

Amount of space currently used by the storage pool.

### Used (%)

Percentage of the total storage pool size that is currently being used.

### Threshold (%)

Storage pool usage percentage threshold. When the **Used (%)** reaches this threshold, SAN File System will send an alert.

### Volumes

Number of volumes in the associated storage pool.

---

## Add Volumes to a Storage Pool - Verify Settings panel

This panel (the final step in the **Add Volumes to a Storage Pool** wizard) lets you confirm your selections before the volumes are added.

All of the displayed fields on this verification panel are read-only. To make changes, return to any of the previous panels in the wizard.

### Attributes

The New Volumes table lists the size of each volume in megabytes, the LUN ID associated with the volume, as well as the following information:

#### Volume Name

The name that will apply to the new volume.

#### Initial Status

The status of the volume can be **Active** or **Suspended**.

**Note:** If volumes are in a suspended state, data cannot be written to them. You can change a volume to the active state by going back to the previous step of the wizard (**Add Volumes**) and select the checkbox to **Initially activate volumes**.

---

## Add Volumes to a Storage Pool - Volume Settings panel

This panel (the fourth step in the **Add Volumes to a Storage Pool** wizard) allows you to specify settings to access information about logical unit numbers (LUNs) that are available to be added to the storage pool as volumes.

### Attributes

#### Volume Name Prefix

Specifies the prefix used for generating the volume name. The format of a generated volume name is:

*<prefix>-<storage pool name>-<integer starting at 0 and increased by 1 for each volume name>* For example, if you enter the prefix "eng," your volumes might become eng-MainLabPool-0, eng-MainLabPool-1, eng-EngPrivatePool-0 and so forth.

**Tip:** You can rename the volumes later.

#### Initially activate volumes

Adds the volumes and activates them when added. If a volume is not activated, it is in a suspended state, and data cannot be written to it.

#### Forced Addition

Forces the metadata server to add the volume and write a new label to the volume if the volume already has a valid SAN File System label. When not selected, this setting ensures that the volumes are not currently assigned in the associated cluster.

---

## Administration Properties – Details panel

Use this panel to view the current settings of administration properties.

The administration details table provides the following Remote Service Adapter (RSA) and Lightweight Directory Access Protocol (LDAP) information:



## **Attributes**

### **RSA User ID**

The RSA user ID that is used to access information from the engines.

### **RSA Password**

The password corresponding to the RSA user ID.

### **LDAP User ID**

The administrative LDAP user ID used to make authentication queries.

### **LDAP Password**

The password corresponding to the administrative LDAP user ID.

### **LDAP Server IP Address**

The IP address of the LDAP server.

### **LDAP Session Authorization Timeout Limit (seconds)**

The maximum age, in seconds, of items in LDAP cache.

### **User Attribute Name**

The LDAP attribute containing the login name.

### **Role Attribute Name**

The LDAP attribute containing the name of the role.

### **Role Membership Attribute Name**

The LDAP attribute containing the role's members.

### **Roles Distinguished Name**

The base distinguished name (DN) where roles are located in the LDAP server.

---

## **Administration Properties – LDAP Access panel**

Use this panel to modify Lightweight Directory Access Protocol (LDAP) configuration settings on the metadata server.

## **Attributes**

Change the values in the fields to update LDAP configuration settings.

**Restriction:** You must change LDAP settings on the metadata server using a single submission in the console. You can then modify the same settings on the LDAP server.

### **LDAP User ID**

The LDAP user ID is used to resolve the user names to user distinguished names (DNs) that are needed to validate the SAN File System user. It can be the LDAP administrator ID or any valid user ID in the LDAP server that has read access to the users and roles.

### **LDAP Password**

The password that corresponds to the LDAP user ID.

### **Verify LDAP Password**

The LDAP password that if you modified you must enter again to verify that you entered it as intended.

### **LDAP Server IP Address**

The IP address of the LDAP server.

**Roles Distinguished Name**

The base distinguished name (DN) where roles are located in the LDAP server. In LDAP, distinguished names are paths of name-value pairs separated by commas, for example, ou=Roles,o=IBM,c=US. To specify more than one role name, separate each name with a comma.

**User Attribute Name**

The LDAP attribute containing the login name.

**Role Attribute Name**

The LDAP attribute containing the name of the role.

**Role Membership Attribute Name**

The LDAP attribute containing the members of the role.

**LDAP Session Authorization Timeout Limit**

The maximum age, in seconds, of items in LDAP cache.

---

## Administrative Properties – System Access panel

Use this panel to update a Remote Service Adapter (RSA) user ID and password, which are used to access information from the engines. The values must match those in the RSA system.

**Attributes**

Change the fields to specify a different RSA user ID and password.

**RSA User ID**

The RSA user ID.

**RSA Password**

The password that corresponds to the RSA user ID.

**Verify RSA Password**

The RSA password that you must verify to ensure that you entered it as intended.

---

## Administrative Log panel

This panel enables you to monitor SAN File System administrative processes that are related to the administrative agent.

**Fields**

Using the following panel fields, you can set certain parameters to specify which messages the administrative log retrieves and refresh the log based on those settings.

**Order and date of log messages**

Specifies the way in which you can limit which messages are retrieved. There are two ways in which you can specify which messages to retrieve based on when the messages were issued:

**Most recent first**

Retrieves the most recent rows in the log.

**Most recent last, beginning on this date:**

Retrieves rows in the log beginning with the date specified.

**Levels** Filters the messages that are displayed, based on message severity level.

**Refresh**

Reloads the log based on the adjacent log-retrieval settings.

**Older Messages**

Selects rows in the log, downward, relative to the current view.

**Newer Messages**

Selects rows in the log, upward, relative to the current view.

**Actions**

**Help** Displays explanation and action information for a selected message.

**Clear Log**

Deletes all messages in the log.

**Fields**

You can select a log entry on which to perform one of the actions described above. The administrative log table fields provide the following details:

**Message ID**

Each message follows this convention:

“XXX” Component

“YY” Subcomponent

“nnnn”

Message code

“Z” Severity (Severe, Error, Warning, or Information level)

**Date and Time**

The system (not local) time that the message was issued.

**Level** The severity level of the message. Four levels exist:

-  Severe
-  Error
-  Warning
-  Information

**Server** The metadata server name that is the source of the message.

**Message**

The text of the message.

---

## Attach Fileset panel

Use this panel to attach a fileset using a different directory name.

**Actions****Attach Point**

Path and directory location of the fileset within the global namespace.

**Fields****Existing Directory Path**

Specifies the existing directory path (relative to the root of the global

namespace and without the / prefix) at which to attach the fileset. This directory must already exist on the client side. The root of the global namespace must be included in the directory path. Use only forward slashes (/) in the directory path for delimiters.

#### **New Directory Name**

Specifies the directory name for the root of the fileset directory tree. The directory name must not already exist on the client side. This directory can be up to 256 characters in length and must not contain backslash (\) or forward slash (/) characters as delimiters.

**Note:** You must enter the new directory name when the fileset is created, moved or reattached.

---

## **Audit Log panel**

This panel enables you to monitor SAN File System status.

### **Fields**

Using the following panel fields, you can set certain parameters to specify which messages the audit log retrieves and refresh the log based on those settings.

#### **Order and date of log messages**

Specifies the way in which you can limit which messages are retrieved. There are two ways in which you can specify which messages to retrieve, based upon when the messages were issued:

##### **Most recent first**

Retrieves the most recent rows in the log.

##### **Most recent last, beginning on this date:**

Retrieves rows in the log beginning with the date specified.

**Levels** Filters the messages displayed, based on message severity level.

#### **Refresh**

Reloads the log based on the adjacent log-retrieval settings.

#### **Older Messages**

Selects rows in the log, downward, relative to its current view.

#### **Newer Messages**

Selects rows in the log, upward, relative to its current view.

### **Actions**

**Help** Displays explanation and action information for a selected message.

#### **Clear Log**

Deletes all messages in the log.

### **Fields**

You can select a log entry on which to perform one of the actions described above. The audit log table fields provide the following details:

#### **Message ID**

Each message follows this convention:

“XXX” Component

"YY" Subcomponent

"nnnn"  
Message code

"Z" Severity (Severe, Error, Warning, or Information level)

**Date and Time**

The system (not local) time that the message was issued.

**Level** The severity level of the message. Four possible levels exist:

-  Severe
-  Error
-  Warning
-  Information

**Server** The metadata server name that is the source of the message.

**Message**

The text of the message.

---

## Available Data LUNs panel

Use this panel to view and perform one of the actions listed below on the data logical unit numbers (LUNs) designated for use by SAN File System, but not currently assigned to a user storage pool.

Available data LUNs are the writable data LUNs that can be seen by SAN File System, but have not yet been assigned to a user storage pool.

**Tip:** To view all data LUNs that SAN File System can access, whether or not they have been assigned to a user storage pool, select the **Data LUNs** panel.

### Fields

Using the following panel fields, specify settings from which to create the available data LUNs table and press Refresh.

**Client** Selects a client that can see the data LUNs.

**Method of fetching the LUN information**

Selects the method to use in order to fetch the LUN information from cache or to rediscover the LUNs.

**Refresh**

Refreshes the Available Data LUNs table based on the selected client and fetch method.

### Actions

**Add as volumes to a user pool**

Takes you to a wizard for adding a volume to a storage pool.

**Tip:** When entering the Add Volumes wizard from the **Available Data LUNs** panel, your chosen LUNs are preselected in the wizard.

**Clients that can see the LUN**

Lists the clients that can see the selected data LUN.

**Details**

Displays additional LUN details, such as OS device path and sector size.

**Fields**

The available data LUNs table allows you to select one or more LUNs to add as volumes to a user storage pool or perform one of the other actions described above. In addition, the table fields provide the following information:

**LUN ID**

Identifier of the LUN. When you click on a LUN ID in the table, you link to additional details about the associated LUN.

**Volume Name**

The volume name, if the LUN is a volume assigned by SAN File System. The system displays a dash (-) if the LUN has not been assigned as a volume.

**State** The state of the LUN. The LUN state can be one of the following:

**Assigned**

Indicates that the LUN is already assigned to a SAN File System storage pool as a usable volume.

**Available**

Indicates the usable LUNs that can be seen by SAN File System, but are not currently assigned to a storage pool.

**Error** Indicates that there was an error in determining the properties of the LUN.

**Unknown**

Indicates that the metadata server is not running, so a LUN state could not be determined (that is, the property could not be queried).

**Unusable**

Indicates the LUNs that are unsuitable to be a SAN File System volume. Possible reasons that a LUN might be unusable include the following:

- The LUN cannot process read or write operations.
- There is inconsistent availability of a LUN from all servers in the cluster.
- There are disk or driver configuration errors.
- A LUN type is not recognizable to the system.

**Note:** The unusable state takes precedence over the assigned or available states.

**Size (MB)**

The total size of the LUN in megabytes.

**Vendor**

The name of the vendor of the storage device associated with the LUN.

**Product**

The product name of the storage device that is associated with the LUN. The system displays the following IBM products in addition to other companies' products:

<b>2105F20</b>	ESS F20
<b>2105800</b>	ESS 800
<b>2145</b>	SAN Volume Controller and SAN Integration Server
<b>3552</b>	FAStT 5000
<b>1722</b>	FAStT 600
<b>1742</b>	FAStT 700 and 900

---

## Available Metadata LUNs panel

Use this panel to view and perform actions on the metadata logical unit numbers (LUNs) designated for use by SAN File System, but not currently assigned to the system storage pool.

Available metadata LUNs are the writable LUNs that can be seen by SAN File System, but have not yet been assigned to the system storage pool.

**Tip:** To view all metadata LUNs that SAN File System can access, whether or not they have been assigned to the system storage pool, select the **Metadata LUNs** panel.

### Actions

#### Add as volumes to the system pool

Takes you to a wizard so you can add a volume to a storage pool.

**Tip:** When entering the Add Volumes wizard from the **Available Metadata LUNs** panel, your chosen LUNs are preselected in the wizard.

#### Rediscover LUNs

Refreshes the LUN information from the LUNs, not from the cache.

### Details

Displays additional LUN details, such as the volume name, state, OS device path, and sector size.

### Fields

The available metadata LUNs table allows you to select one or more LUNs to add as volumes to the system storage pool. In addition, the table displays the following information:

#### LUN ID

The LUN identifier. When you click on a LUN ID in the table, you link to additional details about the associated LUN.

#### Size (MB)

The total size of the LUN in megabytes.

#### Vendor

The name of the vendor of the storage device associated with the LUN.

#### Product

The product name of the storage device associated with the LUN. The system displays the following IBM products in addition to other companies' products:

- 2105F20  
ESS F20
- 2105800  
ESS 800
- 2145 SAN Volume Controller and SAN Integration Server
- 3552 FAStT 5000
- 1722 FAStT 600
- 1742 FAStT 700 and 900

#### Storage Device WWNN

The world-wide node name of the hosting storage device (for example, ESS or FAStT). This information is helpful when you need to ensure that all LUNs in a storage pool are from the same storage device.

---

## Change Power State of Engine panel

Use this panel to verify the engine or engines selected for a power state change, and then choose a change method.

#### Attributes

##### Engines Selected for Power State Change

Read-only list of the selected engines that are now targeted for power state change.

#### Actions

##### Power State Change Method

Select from one of the five listed methods for changing the engine's power state.

**Note:** For "Schedule Power On Or Off," select the radio button next to that method, and then select either **On** or **Off**.

##### Time settings

Select the year, month, day, and hour when you want the power state change to take effect.

**Note:** The Current ASM field is a read-only field. This is the current time on the local clock for the Advanced Systems Management device's local clock. This time reference must be used to schedule the "power-off," operation and is independent of the time on the engine.

---

## Change State of Cluster panel

Use this panel to change the state of the cluster in one of four ways: bring online, fully or partly quiesce, or take offline. The cluster state change prepares both clients and servers for volume-level backup.

#### Fields

##### Resume to Online

Resumes the cluster if it is in a quiescent or offline state. (Default choice if the cluster is fully quiescent, partly quiescent, or offline.)



### Partly Quiesce

Only quiesces the metadata activity. Allows existing client file input/output (I/O) activity to continue (used for “dirty backups”). (Default choice if the cluster is online.)

### Fully Quiesce

Quiesces all metadata and client activity (used for “clean backups”). All data and metadata is flushed to disk.

**Attention:** Moving to this state could cause a major delay to users that have current client sessions.

### Take Offline

Suspends all client sessions.

**Attention:** Moving to this state could cause a major delay to users that have current client sessions.

---

## Check Metadata panel

Use this panel to check the integrity of SAN File System metadata. You also have the option of repairing the metadata.

### Attributes

#### Repair as necessary

Enables the check procedure to repair metadata as needed.

**Note:** The cluster may need to be taken offline and require manual intervention to repair metadata.

### Metadata Type

Specify which type of metadata to check, repair, or both:

- Content and Structure (The default is enabled.)
- Content Only
- Structure Only

### Metadata Location

Select which storage pool and fileset combination you want the check to run against:

- System Pool and All Filesets (The default is enabled.)
- System Pool Only
- All Filesets
- Specific Filesets

### Fields

If you choose to run the metadata check only against specific filesets (as specified in the **Metadata Location** setting), you need to select one or more filesets from the Filesets table. The table lists the following details:

**Name** Name of the associated fileset.

**State** Two fileset states are possible:

- Attached
- Detached

**Quota (MB)**

The quota for the fileset in megabytes.

**Threshold (%)**

The percentage of the quota that the fileset must reach before an alert is issued.

**Attach Point**

Path and directory location of the fileset within the global namespace.  
(This field displays a dash (-) if the fileset is detached.)

---

## Check Metadata Progress panel

Use this panel to view the progress of the metadata check and repair task. You can also stop the check and repair process from this panel, because the process can be time-consuming.

**Fields****Progress Status Area****Progress text**

Displays one of the following, depending on the check and repair task status:

- “The metadata check is in progress.”
- “The metadata check completed <time stamp>.”
- “The metadata check was stopped <time stamp>.”

**Progress bar**

Animated graphic that continues until the metadata check completes, or is stopped.

**Attributes****Metadata Check Settings****Repair as necessary**

Indicates if the repair operation is enabled. If so, the overall “check metadata” task will repair metadata as needed.

**Metadata Type**

Indicates which type of metadata is being checked, repaired, or both (if the repair setting is enabled).

**Metadata Location**

Indicates which storage pool, fileset, or combination the check is running against.

**Fields****Pushbuttons**

.

**Stop** Stops the check. Does not close the panel.

**Note:** This pushbutton becomes disabled when the operation is complete.

**Close** Closes the panel.

**Note:**

1. The close action does not stop the underlying check if it is currently running.
2. If you close this panel before the check completes, you cannot stop the check process from the SAN File System console. However, you can stop the process from the command line interface (CLI) using the `stopmetadatacheck` command.

---

## Client Sessions panel

Use this panel to grant or revoke root privileges for SAN File System clients, or to view client session properties.

There are two client privilege levels, also called access levels:

**Root** Privileged client-access level.

**Standard**

Normal client-access level. This access level is the default.

**Actions**

**Grant Clients Root Privileges**

Allows you to grant unrestricted access to the selected clients.

**Note:** You can also perform this action from the **Privileged Clients** panel.

**Revoke Clients Root Privileges**

Removes unrestricted access from the selected clients.

**Note:** You can also perform this action from the **Privileged Clients** panel.

**Details**

Displays additional detailed properties of the selected client, including transactions and locks.

**Fields**

You can select a client for which to grant root privileges, revoke root privileges, or view details. The client sessions table lists the following information:

**Client** The name of each client. This name is provided on the client side.

**Note:** It is not an IP host name.

**Session ID**

Session identifier. When you click on a Session ID in the table, you link to additional details about the selected client.

**Tip:** A single client can have multiple, simultaneous sessions — one per metadata server.

**State** The client state. The client session states can be one of the following:

**Expired**

Lease is expired.

**Current**

Lease is currently valid.

A client might not be connected, in which case, neither state applies.

**Server** The name of the metadata server that is part of the client session.

**Renewals**

The count of session renewals that increments once per renewal. The renewal gives you an indication of the length of each client session.

**Client Privilege**

The client root privileges, which is one of these options:

- Root (privileged client)
- Standard (normal client)

---

## Clients that can see the LUN panel

Use this panel to view a list of clients that can see the selected LUN.

**Attributes**

The panel lists all the clients that can currently see the LUN.

---

## Cluster panel

Use this panel to perform an action on the SAN File System cluster, or to view its properties.

**Actions**

**Properties**

- Displays detailed properties of the SAN File System cluster
- Provides a way for you to view tuning settings
- Provides a way for you to modify the storage pool-space-reclamation interval

**Start Online**

Starts the cluster in an online state.

**Note:** Starting the cluster could take some time.

**Start Offline**

Starts the cluster in an offline state.

**Note:** Starting the cluster could take some time.

**Change State**

Lets you change the state of the cluster (bring the cluster online, fully or partly quiesce the cluster, or take the cluster offline). Cluster state change prepares both clients and metadata servers for volume-level backup.

**Stop**

Lets you gracefully stop the metadata servers in the cluster. The metadata server and their associated states, roles, and filesets are displayed for your review and confirmation before the stop operation begins.

**Commit Software Upgrade**

Upgrades the currently operational software version on the cluster.

**Requirements:**

1. Cluster upgrades can occur only after each individual metadata server has been upgraded to the same new

software version. Individual metadata server upgrades must be done at the Administrative command-line interface.

2. The cluster upgrade can only be run on the master metadata server.
3. The cluster does not need to be stopped for the upgrade.

The Cluster table lists these cluster properties:

- Name
- System identification number
- Current state
- The date and time of the last state and target state changes

In addition, the table provides the following information:

#### **Attributes**

##### **Target State**

The state to which the cluster is progressing. For example, if the system is booting up, the pending target state might be "Online."

##### **Servers**

The total number of metadata servers in the cluster.

##### **Active Servers**

The number of metadata servers in an active state (online, offline or quiesced).

##### **Privileged Clients**

The number of clients that have been granted root privileges.

##### **Software Version**

Current software version on the cluster. It might not have been committed.

##### **Committed Software Version**

The currently committed software version on the cluster. This attribute does not indicate whether the commit has finished because it only updates the metadata structures on-demand, when they are accessed.

**Tip:** Typically, if an upgrade is in progress, the committed software version is an earlier version than the pending cluster software version.

##### **Last Upgrade Commit**

The date and time that last software commit was initiated.

##### **Upgrade Commit Status**

Indicates the status of the upgrade:

###### **In progress**

The commit is ongoing. A commit can take a long time to complete because it is done on-demand.

**Note:** Future upgrades cannot be started if the software commit status is in progress.

###### **Not in progress**

Either a commit has never been done, or one has completed.

### Installation Date

The date and time that the global fileset was created (that is, when SAN File System was installed).

---

## Cluster Log panel

This panel helps you to monitor the servers in the SAN File System cluster.

### Fields

Using the following panel fields, you can set certain parameters to limit which messages the cluster log retrieves and refresh the log based on those settings.

#### Order and date of log messages

Specifies the way in which you can limit which messages are retrieved. There are two ways that you can specify which messages to retrieve, based upon when the messages were issued:

##### Most recent first

Retrieves the most recent rows in the log.

##### Most recent last, beginning on this date:

Retrieves rows in the log beginning with the date specified.

**Levels** Filters the messages that are displayed, based on message severity level.

**Types** Sets the filter to display all messages or events only.

**Note:** An event is a significant or major message that can often, but not necessarily, trigger an alert.

### Refresh

Reloads the log based on the log-retrieval settings.

### Older Messages

Retrieves rows in the log, downward, relative to the current view.

### Newer Messages

Retrieves rows in the log, upward, relative to the current view.

### Actions

**Help** Displays explanation and action information for a selected message.

### Clear Log

Deletes all messages in the log.

### Fields

You can select a log entry on which to perform one of the actions described above. The cluster log table fields provide the following details:

#### Message ID

Each message follows this convention:

"XXX" Component

"YY" Subcomponent

"nnnn"  
Message code

"Z" Severity (Severe, Error, Warning, or Information level)

**Date and Time**

The system (not local) time that the message was issued.

**Level** The severity level of the message. Four possible levels exist:

-  Severe
-  Error
-  Warning
-  Information

**Type** The message types fall into one of three categories:

**Normal**

Messages that are not events.

**Event** A significant or major message can often, but not necessarily, trigger an alert.

**Note** Other message types such as “audit” or “trace.”

**Server** The metadata server name that is the source of the message.

**Message**

The text of the message.

---

## Cluster Properties – Details panel

Use this panel to view information about the SAN File System cluster.

The cluster details table lists these cluster properties:

- Name
- System identification number
- Current state
- The date and time of the last state and target state changes.

In addition, the cluster details table provides the following information:

**Attributes****Target State**

The state to which the cluster is progressing. For example, if the system is booting up, the pending target state might be “online.”

**Servers**

The total number of servers in the cluster.

**Active Servers**

The number of servers in an active state (online, offline or quiesced).

**Privileged Clients**

The number of clients that have been granted root privileges.

**Software Version**

The next software version to which the cluster is set to move.

**Committed Software Version**

The current software version on the cluster.

**Tip:** If an upgrade is in progress, the Operational Software Version is an earlier version than the Pending Software Version.

**Last Upgrade Commit**

The date and time of the last upgrade to the cluster software.

**Upgrade Commit Status**

Indicates the status of an upgrade. The status can be “In progress,” or “Not in progress.”

**Installation Date**

The date and time that the global (root) fileset was created (that is, when SAN File System was installed).

**Language**

The language in which the metadata server runs.

**Pool Space Reclamation Interval (minutes)**

The interval that specifies how often SAN File System reclaims disk space allocated to storage pools. You can reset this parameter on the **Cluster Properties - Tuning** panel.

**Client Lease Period (seconds)**

The amount of time a metadata server grants a client access to a lock. The range is from 10 to 120 seconds.

**Note:** The metadata server also applies the multiplier to this value before the lease expires.

**Client Lease Period Multiplier**

A value that directly interacts with the lease period to accommodate possible networking delays between when a message gets sent from the client and when it is received by the metadata server. The range is from 0 to 4. A value of 0 means “disabled.”

**Master Server Buffer Size (pages)**

The number of additional pages that are to serve as a space buffer on the master metadata server. The range is from 2 048 to 8 192 pages. The installation default is 2 048.

**Subordinate Server Buffer Size (pages)**

The number of additional pages that are to serve as a space buffer on the subordinate metadata servers in the cluster. The range is from 30 000 to 250 000 pages. The installation default is 30 000.

**Maximum Network Heartbeats**

The maximum number of heartbeats that can be missed before declaring a path failure on a subsequent missed heartbeat. A metadata server is ejected from the cluster if it exceeds any of the heartbeat thresholds. The range is 1 to 100.

**Note:** When the maximum setting is reached, if the metadata server was already up and running, the metadata server goes into a “Not running” state and remains that way until repaired. If the metadata server was not up and running, the ejected metadata server restarts and rejoins the cluster.

**Network Heartbeat Interval (msec)**

The interval between sending heartbeats. The range is 20 to 10 000.



**Within-Cluster Timeout (msec)**

The timeout period for intracluster communications. The range 500 000 to 10 000 000.

**Retries to Client**

The maximum times the metadata server attempts to contact the client before declaring the client's session lease to be expired. The range is 1 to 100. The installation default is 5.

---

## Cluster Properties – Localization panel

Use this panel to set the language in which the metadata server runs for SAN File System.

**Fields****Language**

Selects the language in which the metadata server runs. Choose en\_US.utf8 for United States English, or ja\_JP.utf8 for Japanese.

---

## Cluster Properties – Tuning panel

Use this panel to set the storage pool-space-reclamation-interval for SAN File System.

**Fields****Storage Pool Space Reclamation Interval checkbox**

Indicates, when the checkbox is checked, that you would like to set how often SAN File System reclaims disk space allocated to storage pools.

**Minutes fields**

Specifies the interval, in minutes, if the setting is enabled. The range is from 1 to 1 440. The default is 60.

---

## Collect Diagnostic Data panel

Use this panel to view detailed information on engines to assist with monitoring and troubleshooting.

You can select one or more engines on which to view additional details.

**Note:** The diagnostic data collection might take several minutes to complete.

The Engines table lists the following information for each engine:

**Fields****Engine IP**

The IP address of the associated engine, which links to its summary notebook panel. The engine summary panel also provides status on temperature, voltage and fans.

**Server** The metadata server residing on the associated engine. This column also identifies the master-subordinate relationship among the metadata servers.

**Note:** The master metadata server will have the  icon next to it.

### Server State

The state of the engine's associated metadata server will be one of the following:

- Not running
- Failed initialization
- Initializing
- Not added
- Joining
- Offline
- Fully quiescent
- Partly quiescent
- Online
- Unknown

---

## Collect Diagnostic Data Progress panel

Use this panel to view the progress of the engine diagnostic data collection task.

### Fields

#### Progress Status Area

##### Progress text

Displays one of the following, depending on the data collection task status:

- "The diagnostic data collection is in progress."
- "The diagnostic data collection completed <time stamp>."
- "The diagnostic data collection ended at <time stamp> and did not complete."

**Note:** This last message appears if an error is encountered.

##### Progress bar

Animated graphic that continues until data collection completes for all engines.

### Attributes

The Collections on Each Engine table specifies where the data collections for the listed engines reside.

#### Diagnostic Data

Name of the output directory where the data is located.

#### Status Log

Name of the output directory where errors associated with the diagnostic data collection task are logged.

### Attributes

The Collection Status table lists the engines on which the data was collected, and the collection status for each.

#### Engine IP

IP address of the associated engine.

**Server**

The metadata server residing on the associated engine.

**Status** The data collection status for the associated engine. Four values are possible:

- Not Started
- In Progress
- Completed
- Failed

**Fields****Pushbuttons**

**Close** Closes the panel.

**Note:** The close action does not stop the underlying data collection if it is currently running.

---

## Create a Fileset panel

Use this panel to create a new fileset in SAN File System.

**Attributes**

**Name** The name for the fileset.

**Description**

A description for the fileset.

**Server Assignment Method**

The method to have the system automatically assign the fileset to a metadata server or to manually assign the fileset to a specific metadata server. To manually select a server, use the drop-down list to view all of the servers known to SAN File System, regardless of their current state.

The metadata server that you select to run the associated fileset must meet the following conditions:

- The target metadata server must be part of the cluster.
- The master metadata server must not be down.
- The global fileset must always be bound to the master metadata server.

**Quota Options**

The quota type, size, and alert triggers.

**Quota Type**

The quota type, which is one of the following:

**Soft quota**

Only issues a warning alert. This quota type allows the use of additional storage even when this level has been reached. This is the default setting.

**Hard quota**

Issues an error level alert and prohibits the use of additional storage when this level has been reached.

**Quota Size**

The quota size and unit of measurement (MB, GB, TB, or PB) for the fileset.

**Restriction:** The size cannot exceed 1 PB.

**Enable usage alerts checkbox**

The manner in which you choose whether usage-based alerts are generated. Select the checkbox to enable quota usage alerts. Clear the checkbox to disable quota usage alerts. If enabled, usage-based alerts (warning level) are generated per the usage threshold setting.

**Usage Threshold**

The percentage of the fileset quota that, when reached, issues an alert. This field is required if the **Enable usage alerts** checkbox is enabled.

**Attach Point**

The path and directory location of the fileset within the global namespace.

**Existing Directory Path**

The existing directory path (relative to the root of the global namespace and without the slash (/) prefix) at which to attach the fileset. This directory must already exist on the client side. The root of the global namespace must be included in the directory path. Use only forward slashes (/) in the directory path for delimiters.

**New Directory Name**

The directory name for the root of the fileset directory tree. The directory name must not already exist on the client side. This directory can be up to 256 characters in length and must not contain backslash (\) or forward slash (/) characters as delimiters.

**Requirement:** You must enter the new directory name when the fileset is created, moved or reattached.

---

## Create FlashCopy Images of Filesets - Introduction panel

Use the **Create FlashCopy Images of Filesets** wizard to create new SAN File System FlashCopy images. This three-step wizard will allow you to select the fileset from which to create the FlashCopy image, set properties for the FlashCopy image, and verify your settings before it is created.

You should be aware of the following rules regarding FlashCopy image creation:

1. After it is created, you cannot change the name, description, or the directory name of a FlashCopy image.
2. There is a limit of 32 FlashCopy images per fileset.
3. A fileset may be attached or detached when the FlashCopy image is created.
4. The **Next** button in the wizard will be disabled if any failures occur during the image creation process.

---

## Create FlashCopy Images of Filesets - Select Filesets panel

This panel (the first step in the **Create FlashCopy Images of Filesets** wizard) allows you to choose one or more filesets for which to create the FlashCopy image.

If you entered the wizard from the **Filesets** panel, the fileset or filesets that you chose before will be preselected.

## Fields

In addition to its description, the table provides the following information about each fileset:

**Name** The fileset name. Clicking the fileset name in the table links to additional details about the selected fileset.

**State** Two fileset states are possible:

- Attached
- Detached

**Server** The metadata server that is manually assigned to the fileset.

### Assigned Server

The metadata server that is manually assigned to serve the fileset.

### Quota (MB)

The quota for the fileset, displayed in megabytes.

### Quota Used (%)

System percentage used by the fileset.

### Allocated (MB)

The number of megabytes used by the fileset.

### Children

Number of subfilesets.

### Most Recent Image

Date and time that the last FlashCopy image was taken for the associated fileset.

---

## Create FlashCopy Images of Filesets - Set Properties panel

This panel (the second step in the **Create FlashCopy Images of Filesets** wizard) allows you to add detailed information about the FlashCopy image, including the image name, directory name and description.

### Attributes

#### Force Image Creation

When selected, this setting deletes the oldest image to make room for a new one, if necessary.

**Note:** There is a limit of 32 FlashCopy images per fileset.

### Attributes

In addition to allowing you to enter an optional detailed description for the FlashCopy image, the **FlashCopy Images Settings per Fileset** section requires you to enter the following:

#### FlashCopy Image Name

Will be prefilled with the same name used in the Directory Name field. However, you can change the name if you want.

#### Directory Name

Will be prefilled with a name according to the following convention: "Image-**<sequence number>**." However, you can change the name if you want.

---

## Create FlashCopy Images of Filesets - Verify Settings panel

This panel (the final step in the **Create FlashCopy Images of Filesets** wizard) allows you to confirm your selected configuration for the new FlashCopy image before it is created.

All of the displayed fields on this panel are read-only. To make changes, return to any of the previous panels in the wizard.

### Attributes

The **FlashCopy Images to Be Created** table lists the fileset, directory name, and description (if entered) for the FlashCopy image. It also provides the following:

**Name** This name will be applied to the new image that you are creating.

---

## Create a Policy - Add Rules panel

This panel (the second step in the **Create a Policy** wizard) allows you to add rules to a newly-created policy. You can add any number of file-placement rules to the policy.

All date and time attributes in these rules are based in coordinated universal time (UTC). Therefore, any time and date entry is assumed to be in universal time. Also, functions (for example, "dayofweek") are based on UTC.

### Fields

#### Rules Description

This optional description is added as a comment statement before each rule.

#### Action: Storage Pool Assignment

Assign the policy rule to a storage pool. The drop-down list presents all of the existing storage pools, except the system storage pool, which is reserved for system data.

#### Conditions

After the policy is activated, if one or more of the enabled conditions matches a new file being stored, it will be assigned to the storage pool that was selected from the **Storage Pool Assignment** drop-down list. Select one or more of the following conditions for the file placement rule:

##### File name

Places a condition on the file name. You can place a file based on a name with specified characters appearing at the beginning ("starts with"), end ("ends with"), or anywhere within it, including the beginning or the end ("contains"). You may also select the "is" option if you want to specify an exact file name, or "is not" if you are sure of what you do not want the name to contain for files assigned to the associated storage pool.

##### Note:

1. You can use a % wildcard to represent one or more characters and use the \_ wildcard to represent a single character.
2. Allowed characters include letters, numerals, %.
3. You can only specify one file name type per rule.

4. Multiple listings are not allowed.

**File created**

Places a condition on the year, month, day and hour the file was created.

**Fileset** Places a condition based on the existing fileset to which the file is assigned.

**Group ID (UNIX only)**

For UNIX-based files. This condition only takes a numeric value. Valid range is 0 – 4 294 967 295 (10–digit maximum).

**User ID (UNIX only)**

For UNIX-based files. This condition only takes a numeric value. Valid range is 0 – 4 294 967 295 (10–digit maximum).

**New Rule pushbutton**

Clicking **New Rule** saves the rule that you just created and clears the panel to build another rule.

**Note:**

1. You can click **Back** to go back to previous steps, but you cannot go back to previous rules.
2. You can advance to the next step (the rules editor), and bypass the **Add Rules** panel by clicking **Next**, regardless of whether or not any rules have been set. The rules editor allows you to perform alternate actions, such as pasting in a rule set.

---

## Create a Policy - Edit Rules panel

The **Edit Rules** panel of the **Create a Policy** wizard allows you to edit rules of a newly created or cloned policy.

This panel (the final step in the wizard) allows you to edit rules of a newly created policy or cloned policy.

**Note:** All date and time attributes in these rules are based in coordinated universal time (UTC). Therefore, any time and date entry is assumed to be in universal time. Also, functions (for example, “dayofweek”) are based on UTC.

**Attributes****Policy Settings**

Read-only fields indicating the assigned **Name** and **Description** of the policy.

**Attributes**

**Rules** Editable text area, for making changes to the rules. You can perform basic editing tasks, such as cut, copy, and paste.

---

## Create a Policy - High-Level Settings panel

This panel (the first step in the **Create a Policy** wizard) allows you to enter a name and description for the policy, as well as select a policy creation method.

### Attributes

**Name** Enter a name for the policy (required).

### Description

Enter an optional detailed description for the policy.

### Attributes

### Policy Creation Method

#### New Policy

If you choose to create a new policy, another panel displays for specifying the policy rules.

#### Clone Policy

If you choose to clone a policy, you must also choose an **Existing Policy** from the drop-down list.

---

## Create a Policy - Introduction panel

Use this wizard to create new file-placement policies, or clone existing policies in SAN File System. This three-step wizard (two steps if you are cloning) will allow you to add and edit rules to define the policy.

The **Next** button in the wizard will be disabled if any failures occur during the policy creation process.

---

## Create Recovery File panel

Use the **Create Recovery File** panel to create a new disaster recovery file, or to overwrite an existing one.

The recovery file will contain information that can aid in restoring the metadata (that is, the storage pools, volumes, filesets and policies).

### Actions

### Creation Method

**Create** You can choose to create a new recovery file and type the new file name.

### Forced Create

Select an existing file from the drop-down list to overwrite.

---

## Create a Storage Pool - Add Volumes panel

This panel (the third step in the **Create a Storage Pool** wizard) allows you to add volumes to a newly-created storage pool. You can add volumes to the storage pool by selecting from the available logical unit numbers (LUNs).

### Attributes

The available LUNs table displays the available LUNs from which you select to write to the storage pool as volumes. It displays the size of the LUN in megabytes, as well as the following information:



**LUN ID**

The identifier of the LUN. When you click on a LUN ID in the table, you link to additional details about the selected LUN.

**Vendor**

The name of the vendor of the storage device that is associated with the LUN.

**Product**

The product name of the storage device that is associated with the LUN.

**OS Device Path**

The path that points to the SAN Adapter driver.

---

## Create a Storage Pool - Introduction panel

Use the **Create a Storage Pool** wizard to create new user storage pools in SAN File System.

This wizard allows you to set the properties of the new storage pool and add volumes to it.

---

## Create a Storage Pool - Pool Settings panel

This panel (the first step in the **Create a Storage Pool** wizard) allows you to specify a name, description and partition size for the storage pool, as well as a percentage-used-threshold alert.

**Actions**

**Name** Specifies the name for the storage pool. This field is required.

**Description**

Specifies the optional detailed description for the storage pool.

**Logical Partition size**

Specifies the partition size, in megabytes, to use when a fileset allocates space. Select from 16, 64, or 256 MB as the partition size of the new storage pool. The default is 16 MB.

**Restriction:** You cannot change the partition size after it is set.

For normal use, set the partition size to 16MB.

**Tip:** To improve performance, especially for handling extremely large files, specify a larger partition size. When you set a partition size higher than 16 MB, the pools take up more storage.

**Allocation size**

Specifies the allocation size, in kilobytes, by which a file that resides on a storage pool is to be extended. Select from System Automated, 4 KB Fixed, or 128 KB Fixed.

**Restriction:** You cannot change the allocation size after it is set.

**Enable usage alerts checkbox**

Enables an alert when the storage pool partition reaches a particular percentage used.

### Usage Threshold

Specifies the storage pool usage threshold at which point an alert is issued if you enabled usage alerts. The default is 80%.

---

## Create a Storage Pool - Select Client panel

This panel (the second step in the **Create a Storage Pool** wizard) allows you to specify settings to access information about logical unit numbers (LUNs) that are available to be added to the storage pool as volumes.

### Fields

**Client** Selects a client that can see the available LUNs.

### Method of fetching the LUN information

Selects the method to use in order to fetch the LUN information from cache or rediscover the LUNs.

---

## Create a Storage Pool - Verify Settings panel

This panel (the final step in the **Create a Storage Pool** wizard) allows you to confirm your selected configuration for a newly-created storage pool.

### Attributes

The Storage Pool Properties table lists the name, description, and partition size of the new storage pool, as well as the following information:

#### Allocation Size (KB)

The storage pool allocation size that you set in the previous panel of the wizard. (The choices were System Automated, 4 KB Fixed, or 128 KB Fixed.)

**Note:** You cannot change the allocation size after it is set. If you want to change the setting, return to the previous step in the wizard and modify it before clicking **Finish**.

#### Usage Threshold (%)

(If enabled in the **Set Properties** portion of the wizard.) When the amount used of the storage pool reaches this percentage of the total allotted to its partition, the system issues an alert. The default is 80%.

### Attributes

For each volume associated with the newly created storage pool, the New Volumes table lists the size of the volume in megabytes, as well as the LUN ID associated with the volume. The table also lists the following information:

#### Volume Name

The name that will be applied to the new volume.

#### Initial Status

The status can be either **Active** or **Suspended**.

**Note:** If volumes are in a suspended state, data cannot be written to them. You can change a volume to the active state by going back to the previous step of the wizard (**Add Volumes**) and selecting the **Initially activate volumes** checkbox.

---

## Create a Storage Pool - Volume Settings panel

This panel (the fourth step in the **Create a Storage Pool** wizard) allows you to specify settings for the volumes that you are adding to a newly-created storage pool.

### Attributes

#### Volume Name Prefix

Specifies the prefix used for generating the volume name. The format of a generated volume name is:

*<prefix>-<storage pool name>-<integer starting at 0 and increased by 1 for each volume name>*

For example, if you enter the prefix “eng,” your volumes might become eng-MainLabPool-0, eng-MainLabPool-1, eng-EngPrivatePool-0 and so forth.

**Tip:** You can rename the volumes later.

#### Initially activate volumes

Adds the volumes and activates them when added. If you do not initially activate the volumes, they are added in a suspended state, and data cannot be written to them.

#### Forced Addition

Forces the metadata server to add a volume and write a new label to the volume if the volume already has a valid SAN File System label. When not selected, this setting will ensure that the volumes are not currently assigned in the associated cluster.

---

## Data LUNs panel

Use this panel to view detailed information for all data logical unit numbers (LUNs) that can be seen by a specified client. The displayed data LUNs include “read-only,” “assigned” and “available LUNs.”

This panel displays data LUNs that are read-only, those that have been assigned as SAN File System volumes, as well as those that are unassigned, but available.

**Tip:** To view only writable data LUNs that have not yet been assigned to a user storage pool, select the **Available Data LUNs** panel.

### Fields

Using the following panel fields, specify settings from which to create the Data LUNs table and press Refresh.

**Client** Selects a client that can see the data LUNs.

#### Method of fetching the LUN information

Selects the method to use in order to fetch the LUN information from cache or rediscover the LUNs.

#### Refresh

Refreshes the Data LUNs table based on the selected client and fetch method.

## Actions

### Clients that can see the LUN

Lists the clients that can see the selected data LUN.

### Details

Displays additional data LUN details, such as OS device path and sector size.

## Fields

You can select a LUN on which to perform one of the actions described above. The Data LUNs table fields provide the following details:

### LUN ID

The identifier of the LUN. When you click on a LUN ID in the table, you link to additional details about the associated LUN.

### Volume Name

The volume name, if the LUN is a volume assigned by SAN File System. The system displays a dash (-) if the LUN has not been assigned as a volume.

**State** The state of the LUN. The LUN states can be one of the following:

#### Assigned

Indicates that the LUN is already assigned to a SAN File System storage pool as a usable volume.

#### Available

Indicates the usable LUNs that can be seen by SAN File System, but are not currently assigned to a storage pool.

**Error** Indicates that there was an error in determining the properties of the LUN.

#### Unknown

Indicates that the metadata server is not running, so a LUN state could not be determined (that is, the property could not be queried).

#### Unusable

Indicates the LUNs that are unsuitable to be a SAN File System volume. Possible reasons that a LUN might be unusable include the following:

- The LUN does not support read or write operations.
- There is inconsistent availability of a LUN from all servers in the cluster.
- There are disk or driver configuration errors.
- A LUN type is not recognizable to the system.

**Note:** The unusable state takes precedence over the assigned or available states.

### Size (MB)

The total size of the LUN in megabytes.

### Vendor

The vendor of the storage device associated with the LUN.

**Product**

The product name of the storage device that is associated with the LUN. The system displays the following IBM products in addition to other companies' products:

**2105F20**

ESS F20

**2105800**

ESS 800

**2145** SAN Volume Controller and SAN Integration Server

**3552** FAStT 5000

**1722** FAStT 600

**1742** FAStT 700 and 900

---

## Delete Filesets panel

Use this panel to verify the removal of filesets from SAN File System, and choose a method of deletion (regular or forced).

**Fields**

The Filesets Selected for Deletion table displays the name and attach point for the associated fileset or filesets. It also displays the following:

**State** One of two fileset states are possible:

- Attached
- Detached

**Used (MB)**

The number of megabytes used by the fileset.

**Children**

Number of subfilesets. (If there are no subfilesets, "-" is displayed.)

---

## Delete FlashCopy Images panel

Use this panel to review and verify the image or images that you have selected for deletion, as well as select a deletion method (regular or forced).

**Attributes**

The Images Selected for Deletion table displays the name and description of each FlashCopy image so that you can verify your choices before they are deleted.

**Note:** You cannot delete a FlashCopy image that has client activity (session locks open) unless it is a forced delete operation.

The table also lists the following detail for each selected image:

**Name** Name of the FlashCopy image of the associated fileset.

**State** The state of the FlashCopy image. If the FlashCopy image is in a complete state, it is intact. If the FlashCopy image is in an incomplete state, it contains incomplete files caused by a (forced) volume deletion.

**Fileset** Fileset that the image represents.

**Directory Name**

Directory name where the image is located.

**Note:** The FlashCopy image directory is the one below the “.flashcopy” directory of the fileset.

---

## Delete Storage Pools panel

Use this panel to review and verify the storage pool or storage pools that you have selected for deletion.

**Attributes**

The Storage Pools Selected for Deletion table displays the name and description of each storage pool so that you can verify your choices before they are deleted. It also lists the following detail for each selected storage pool:

**Type** Displays the storage pool type or types that you selected for deletion.

**Note:** You can only delete user storage pools. If you attempt to delete the user default or system storage pool, you will get an error message.

**Size (MB)**

The total space available in the associated storage pool.

**Used (MB)**

The amount of space being used in the storage pool.

**Used (%)**

The percentage of the total storage pool size that is currently being used.

**Volumes**

Number of volumes in the associated storage pool.

---

## Detach Filesets panel

Use this panel to verify the filesets to detach, as well as a method of detachment (regular or forced).

**Fields**

In addition to the system name and amount used for the fileset or filesets to be detached, the Detach Fileset table lists the following:

**State** The fileset will be in one of two states:

- Attached
- Detached

**Attach Point**

Path and directory location of the fileset in the global namespace.

---

## Details of Client Session panel

Use this panel to view additional information about client sessions, including transactions and locks.

## Attributes

The Details table lists the client IP address, port, OS Platform, and the following information:

**Client** Identifies each client by its name. This name is given on the client side.

**Note:** It is not an IP host name.

### Session ID

Unique identifier for the client session.

**Note:** A single client may have multiple, simultaneous sessions – one per metadata server.

**State** Two client session states are possible:

#### Expired

Lease is expired.

#### Current

Lease is currently valid.

**Note:** A client might also not be connected, in which case, neither state applies.

### Server Name

Name of the metadata server that is part of the client session.

### Renewals

A count of session renewals that increases once per renewal. The renewal gives you an indication of the length of each client session.

### Last Renewal

Date and time (system, not local) of the last renewal.

### Next Renewal (msec)

Milliseconds until next session renewal is to occur.

### Client Privilege

Refers to the client root privileges, which can be:

- Root (privileged client)
- Standard (normal client)

### Client File System Version

The version includes the client level and the SAN File System Protocol level in parentheses.

### Transactions Started

Transactions started during the client session.

### Transactions Completed

Transactions completed per the client session.

### Session Locks

The current number of session locks for the client session. Clients must acquire session locks to do any operation with a filename.

### Data Locks

The current number of data locks held for the client session. Clients must hold data locks in order to cache data pages, attributes of files, and to cache read-only attributes and contents of directories and links.

### Byte Range Locks

The current number of byte range locks held for the client session. These locks are used to implement lock system calls and have no direct effect on the contents or attributes of SAN File System objects or other file system operations.

---

## Details of Image panel

Use this panel to view additional FlashCopy image attributes not provided on the main **FlashCopy Images** panel.

### Attributes

The Details of Image table displays the description of the FlashCopy image, as well as the following information:

**Name** Name of the FlashCopy image of the associated fileset.

**State** The state of the FlashCopy image. If the FlashCopy image is in a complete state, it is intact and you can revert the fileset to that FlashCopy image. If the FlashCopy image is in an incomplete state, it contains incomplete files caused by a (forced) volume deletion. You cannot revert a fileset to that FlashCopy image.

**Fileset** Fileset that the image represents.

### Fileset state

Two fileset states are possible:

- Attached
- Detached

### Directory Name

Directory name where the image is located.

**Note:** The FlashCopy image directory is the one below the “.flashcopy” directory of the fileset.

### Directory Path

Path where the FlashCopy image directory is located, using the following naming convention:

- the fileset’s attach point
- “/.flashcopy”
- /Directory name

### Date and Time

Indicates when the FlashCopy image was taken.

---

## Details of LUN panel

Use this panel to view additional logical unit numbers (LUN) attributes not provided on the **Data LUNs**, **Metadata LUNs**, **Available Data LUNs**, or **Available Metadata LUNs** panel.

### Attributes

The Details of LUN table displays the following information about the selected LUN:



**LUN ID**

The LUN identifier. When you click on a LUN ID in the table, you link to additional details about the associated LUN.

**Volume Name**

The volume name, if the LUN is a volume assigned by SAN File System. The system displays a dash (-) if the LUN has not been assigned as a volume.

**State** The state of the LUN. The LUN states can be one of the following:

**Assigned**

Indicates that the LUN is already assigned to a SAN File System storage pool as a usable volume.

**Available**

Indicates the usable LUNs that can be seen by SAN File System, but are not currently assigned to a storage pool.

**Error** Indicates that there was an error in determining the properties of the LUN.

**Unknown**

Indicates that the metadata server is not running, so a LUN state could not be determined (that is, the property could not be queried).

**Unusable**

Indicates the LUNs that are unsuitable to be a SAN File System volume. Possible reasons that a LUN might be unusable include the following:

- The LUN does not support read or write operations.
- There is inconsistent availability of a LUN from all servers in the cluster.
- There are disk or driver configuration errors.
- A LUN type is not recognizable to the system.

**Note:** The unusable state takes precedence over the assigned or available states.

**Size (MB)**

The total size of the LUN in megabytes.

**Vendor**

The name of the vendor of the storage device associated with the LUN.

**Product**

The product name of the storage device that is associated with the LUN. The system displays the following IBM products in addition to other companies' products:

**2105F20**

ESS F20

**2105800**

ESS 800

**2145** SAN Volume Controller and SAN Integration Server

**3552** FASTT 5000

**1722** FASTT 600

**Storage Device WWNN**

The unique identifier, also known as the world-wide node name (WWNN), of the storage device. This information is helpful when you need to ensure that all LUNs in a storage pool are from the same storage device.

**OS Device Path**

The path that points to the SAN Adapter driver.

**Tip:** This driver is local to the SAN File System cluster and is not on the LUN storage device.

**Sector Size (Bytes)**

The size of the areas, given in bytes, where blocks of data are physically stored on the LUN's associated storage device.

---

## Disaster Recovery panel

Use this panel to create or delete a disaster recovery file. The recovery file will contain information that can aid in restoring the metadata (that is, the storage pools, volumes, filesets, and policies).

**Attributes****Recovery Directory**

Directory on each metadata server where the recovery files are located. This directory is the same for all servers.

**Build Script**

The script used to generate command files from the metadata dump file.

**Actions**

**Create** Creates a new recovery file or overwrites an existing one.

**Delete** Deletes one or more selected recovery files.

**Fields**

You can select an existing recovery file to delete from the **Recovery Files** table. The table lists the following details:

**Name** Name of the associated recovery file.

**Date and Time**

Indicates when the file was created.

**Size (KB)**

The size of the associated recovery file, in kilobytes.

---

## Download Client Software panel

Use this panel to download and update your local SAN File System client software for Red Hat, Inc. Linux, UNIX, or Windows operating systems.

**Actions****Linux Clients**

Displays a table for Red Hat Linux clients with file names and sizes.

**Name** The alphabetical listing of the client file names, which are linked to the master metadata server's corresponding file. The browser allows you to download the file to your local machine.

**Size** The size of the associated file, in kilobytes.

### UNIX Clients

Displays a table for AIX clients with file names and sizes.

**Name** The alphabetical listing of the client file names, which are linked to the master metadata server's corresponding file. The browser allows you to download the file to your local machine.

**Size** The size of the associated file, in kilobytes.

### Windows Clients

Displays a table for Windows 2000 clients with file names and sizes.

**Name** The alphabetical listing of the client file names, which are linked to the master metadata server's corresponding file. The browser allows you to download the file to your local machine.

**Size** The size of the associated file, in kilobytes.

---

## Engines panel

Use this panel to view engine properties or change the state of an engine.

### Actions

#### Properties

Displays additional detailed properties of the selected engine, including information about temperatures, fans, voltages, timeouts, vital engine data, power, and an engine summary.


#### Change Power State

Allows you to change the power state of the selected engine or engines, as well as choose a change method.

### Fields

#### Engine IP

IP address of the engine, which also links to the engine's **Summary** panel.

**Server** Metadata server associated with the engine. The master metadata server is indicated by the  icon.

#### Boot State

Eight engine boot states are possible:

- Unknown / Power Off
- Before POST (Engine is powered on, but has not started POST.)
- In POST (Engine is in the Power On Self Test)
- Stopped in POST (Error detected.)
- Booted Flash (Engine booted from flash memory.)
- Booting OS
- In OS (The normal state.)

- CPUs Held in Reset

### **Temperature**

Six engine temperature values are possible:

#### **Normal**

The temperature for all engine components that have thresholds are below the warning-level threshold. If no thresholds have been set, a dash (–) is displayed.

#### **Unknown**

The temperature state cannot be determined. Potential reasons include; failure; misconfiguration; or the absence of the RSA card.

#### **Undefined**

Undefined values were returned by the RSA card provider.

#### **Warning**

The temperature of one or more engine components is between the warning level threshold and the soft shutdown level threshold.

**Error** The temperature of one or more engine components is above the soft shutdown level threshold.

**Fans** Four engine fan values are possible:

#### **Normal**

All fans are running in the normal range (15 % or greater speed).

#### **Unknown**

The fan state cannot be determined. Potential reasons include: failure; misconfiguration; or the absence of the RSA card.

#### **Undefined**

Undefined values were returned by the RSA card provider.

#### **Warning**

One or more fans is running below 15% of potential speed.

### **Voltage**

Five engine voltage values are possible:

#### **Normal**

The voltages for all engine components that have thresholds are below the warning-level threshold. If no thresholds have been set, a dash (–) is displayed.

#### **Unknown**

The voltage state cannot be determined. Potential reasons include: failure; misconfiguration; or the absence of the RSA card.

#### **Undefined**

Undefined values were returned by the RSA card provider.

#### **Warning**

The voltage for one or more engine components is either below the low voltage warning threshold, or is above the high voltage warning threshold

---

## **Engines Properties – Fans panel**

Use this panel to view the state and speed status for each fan of the selected engine.

## Fields

**Fan** Each engine has eight fans.

**State** The current state of the associated engine fan. Four engine fan values are possible:

### Normal

All fans are running in the normal range (15 % or greater speed).

### Unknown

The fan state cannot be determined. Potential reasons include: failure; misconfiguration; or the absence of the RSA card.

### Undefined

Undefined values were returned by the RSA card provider.

### Warning

One or more fans is running below 15% of potential speed.

## Speed (%)

Fan speed in relationship to its expected speed. Absolute fan speed (RPMs) will vary by engine depending on the current environmental situation. For example, if one fan goes out, the remaining fans speed up.

---

## Engines Properties – Power panel

Use this panel to view power details, or to change the power state of the associated engine.

### Attributes

**Power** Current power state of the associated engine. The power state will be either “Off” or “On.”

### Power-on Hours

Number of hours the engine has been powered on.

### Restart Count

Number of times the engine has been restarted.

**Note:** This counter is cleared when the Advanced Systems Management (ASM) subsystem is cleared to factory defaults.

### Current ASM Time

This is the current time on the local clock for the Advanced Systems Management device. This time reference is used to schedule a “power-off.” This time is independent of the time on the engine.

### Actions

#### Change Power State pushbutton

Takes you to the **Change Power State of Engine** panel for the single engine in context. This panel enables you to verify the engine selected for a power state change and choose a change method.

---


## Engines Properties – Summary panel

Use this panel to view high-level information about the selected SAN File System engine, including the associated metadata server, IP address, and state.

## Attributes

### Engine IP

IP address of the engine.

**Server** Metadata server associated with the engine. The master metadata server is indicated by the  icon.

### Boot State

Eight engine boot states are possible:

- Unknown / Power Off
- Before POST (Engine is powered on, but has not started POST.)
- In POST (Engine is in the Power On Self Test)
- Stopped in POST (Error detected.)
- Booted Flash (Engine booted from flash memory.)
- Booting OS
- In OS (The normal state.)
- CPUs Held in Reset

### Temperature

Six engine temperature values are possible:

#### Normal

The temperature for all engine components that have thresholds are below the warning-level threshold. If no thresholds have been set, a dash (-) is displayed.

#### Unknown

The temperature state cannot be determined. Potential reasons include; failure; misconfiguration; or the absence of the RSA card.

#### Undefined

Undefined values were returned by the RSA card provider.

#### Warning

The temperature of one or more engine components is between the warning level threshold and the soft shutdown level threshold.

**Error** The temperature of one or more engine components is above the soft shutdown level threshold.

### Voltage

Five engine voltage values are possible:

#### Normal

The voltages for all engine components that have thresholds are below the warning-level threshold. If no thresholds have been set, a dash (-) is displayed.

#### Unknown

The voltage state cannot be determined. Potential reasons include; failure; misconfiguration; or the absence of the RSA card.

#### Undefined

Undefined values were returned by the RSA card provider.

#### Warning

The voltage for one or more engine components is either below the low voltage warning threshold, or is above the high voltage warning threshold

<b>Fans</b>	Four engine fan values are possible:
<b>Normal</b>	All fans are running in the normal range (15 % or greater speed).
<b>Unknown</b>	The fan state cannot be determined. Potential reasons include: failure; misconfiguration; or the absence of the RSA card.
<b>Undefined</b>	Undefined values were returned by the RSA card provider.
<b>Warning</b>	One or more fans is running below 15% of potential speed.

---

## Engines Properties – Temperatures panel

Use this panel to view component temperatures and temperature thresholds for a selected engine.

### Fields

The Component Temperatures table displays the following information:

#### Component

There are five engine temperature components:

- CPU 1
- CPU 2
- DASD 1 – direct access storage device (first hard drive)
- DASD 2 – direct access storage device (second hard drive)
- Ambient – Overall temperature

**State** The temperature state of the associated engine component. Six engine temperature values are possible:

#### Normal

The temperature for all engine components that have thresholds are below the warning-level threshold. If no thresholds have been set, a dash (-) is displayed.

#### Unknown

The temperature state cannot be determined. Potential reasons include; failure; misconfiguration; or the absence of the RSA card.

#### Undefined

Undefined values were returned by the RSA card provider.

#### Warning

The temperature of one or more engine components is between the warning level threshold and the soft shutdown level threshold.

#### Error

The temperature of one or more engine components is above the soft shutdown level threshold.

#### Temperature (°C)

The actual engine temperature of the associated engine component, in degrees Celsius.

### Fields

The Temperature Thresholds table displays the following information:

**Component**

The following four engine components with temperature thresholds are listed:

- CPU 1
- CPU 2
- DASD 1 – direct access storage device (first hard drive)
- DASD 2 – direct access storage device (second hard drive)

**Warning Reset (°C)**

Temperature threshold, for the associated engine component, at which the warning will be reset.

**Warning (°C)**

Temperature threshold for issuing a warning. Temperature threshold, for the associated engine component, at which a warning is issued.

**Soft Shutdown (°C)**

Temperature threshold, for the associated engine component, at which a soft shutdown will occur.

**Note:** A soft, or graceful, shutdown cleanly synchronizes everything before shutting off.

**Hard Shutdown (°C)**

Temperature threshold, for the associated engine component, at which a hard shutdown will occur.

**Note:** A hard, or forced, shutdown stops without a clean synchronization.

---

## Engines Properties – Timeouts panel

Use this panel to view timeout and metadata server restart service details for the associated engine.

**Attributes****POST Server Restart Service (seconds)**

The time, in seconds, that the metadata server restart service waits for Power On Self Test (POST) to complete before sending an alert and auto-restarting the engine. If this feature is disabled, a dash (-) is displayed.

**Loader Server Restart Service (seconds)**

The time, in seconds, that the metadata server restart service waits for the operating system (OS) load to complete before sending an alert and auto-restarting the engine. If this feature is disabled, a dash (-) is displayed.

**OS Server Restart Service Timeout (seconds)**

The total time, in seconds, that the metadata server restart service waits for the OS to respond before sending an alert and auto-restarting the engine. If this feature is disabled, a dash (-) is displayed.

**OS Server Restart Service Check Interval (seconds)**

The time, in seconds, between the OS metadata server restart service timeout checks. If this feature is disabled, a dash (-) is displayed.



**Power Off Delay (seconds)**

The time, in seconds, that the engine waits for the OS to shutdown before turning off the system. The default is 30 seconds. If this feature is disabled, a dash (-) is displayed.

---

## Engines Properties – Vital Engine Data panel

Use this panel to view model, serial number, and firmware details for the associated engine.

**Attributes****Machine Model Number**

Identifies the engine model number.

**Machine Serial Number**

Identifies the engine serial number.

**Firmware Revision**

Identifies the revision numbers and vital product data of the application firmware and the startup read-only memory (ROM) firmware.

**Firmware Revision Date**

The firmware revision date for the logical device.

**Firmware File Name**

The file names and vital product data of the application firmware and the startup ROM firmware.

**Firmware Build ID**

The build IDs and vital product data of the application firmware and the startup ROM firmware.

**Universal Unique Identifier**

Number that uniquely identifies the machine.

---

## Engines Properties – Voltages panel

Use this panel to view component voltages and system board voltage thresholds for the selected engine.

**Fields**

The Component Voltages table displays the following information:

**Component**

There are seven engine voltage components:

- Voltage Regulator Module (VRM) 1 (1.5 Volt)
- Voltage Regulator Module (VRM) 2 (1.5 Volt)
- System Board (SB): 12 Volt
- System Board (SB): 5 Volt
- System Board (SB): 3.3 Volt
- System Board (SB): 2.5 Volt
- System Board (SB): 1.5 Volt

**State** The voltage state of the associated engine component. Five engine voltage values are possible:

**Normal**

The voltages for all engine components that have thresholds are below the warning-level threshold. If no thresholds have been set, a dash (-) is displayed.

**Unknown**

The voltage state cannot be determined. Potential reasons include: failure; misconfiguration; or the absence of the RSA card.

**Undefined**

Undefined values were returned by the RSA card provider.

**Warning**

The voltage for one or more engine components is either below the low voltage warning threshold, or is above the high voltage warning threshold

**Voltage (V)**

The voltage of the associated engine component, listed in volts.

**Fields**

The System Board Voltage Thresholds table displays the following information:

**Component**

The following engine system board voltage components with thresholds are listed:

- 12 Volt
- 5 Volt
- 3.3 Volt
- 2.5 Volt
- 1.5 Volt

**Warning Low**

Component voltage threshold at which a low voltage warning will be issued.

**Warning High**

Component voltage threshold at which a high voltage warning will be issued.

---

## Fileset Properties – Details panel

Use this panel to view additional information about the selected SAN File System fileset.

**Attributes**

The fileset details table lists the fileset name, directory name, directory path, and description. The table also displays the following information:

**State** The state of the fileset. The fileset states can be one of the following:

- Attached
- Detached

**Server Assignment**

Indicates whether the fileset is automatically assigned to a metadata server or if the fileset has manually been assigned a specific metadata server.

**Server** The name of the metadata server that is running the selected fileset.

**Assigned Server**

The metadata server that is manually assigned to serve the fileset.

**Serving State**

Indicates whether the fileset is currently being served. Two states are possible:

- Online

This state means that the metadata server is online and the fileset is attached.

- Offline

This state means that the metadata server is offline.

**Quota Type**

The quota type, which can be soft (alert only) or hard (alert and storage cutoff).

**Quota (MB)**

The fileset's quota, which is displayed in megabytes.

**Tip:** If the quota is not set, the values for the amount of space that is used (Quota Used %) and the threshold (Threshold %) are displayed as 0 and have no meaning. Because the threshold value is based on the amount of space used, if the amount of space used is not defined, any value specified for the threshold value has no meaning and no alert is sent if the displayed value is exceeded.

**Quota Used (%)**

The system percentage used by the fileset.

**Allocated (MB)**

The number of megabytes used by the fileset.

**Threshold (%)**

The percentage used of the quota that the fileset must reach before an alert is issued.

**Attach Point**

The path and directory location of the fileset within the global namespace. This field shows a dash (-) if the fileset is detached.

**Parent** The name of the parent fileset. This field shows a dash (-) if there is no parent fileset.

**Children**

The number of subfilesets.

**FlashCopy Images**

The number of FlashCopy images. The range is from 0 to 32.

**Most Recent Image**

The date and time that the last FlashCopy image was taken for the associated fileset.

---

## Fileset Properties – General Settings panel

Use this panel to change the fileset name, description and metadata server settings.

### Attributes

**Name** The name for the fileset. This name is required.

### Description

Description for the fileset.

### Server Assignment Method

The option of having the system automatically assign the fileset to a metadata server or of having the fileset manually assigned to a specific metadata server. To manually select a different server, use the drop-down list to view all of the servers known to SAN File System, regardless of their current state.

The metadata server must meet the following conditions:

- The target metadata server must be part of the cluster.
- The master metadata server must not be down.
- The global fileset must always be bound to the master metadata server.
- The current metadata server must not be online, fully quiesced, or partly quiesced.

**Tip:** Place the metadata server offline or turn it off.

---

## Fileset Properties – Quota Options panel

Use this panel to change the quota and alert settings.

### Attributes

#### Quota Options

The quota type, size, and alert triggers.

#### Quota Type

The quota type, which is one of the following:

##### Soft quota

Only issues a warning alert. This quota type allows the use of additional storage even when this level has been reached. This is the default setting.

##### Hard quota

Issues an error level alert and prohibits the use of additional storage when this level has been reached.

#### Quota Size

The quota size and unit of measurement (MB, GB, TB, or PB) for the fileset.

**Restriction:** The size cannot exceed 1 PB.

#### Enable usage alerts checkbox

The manner in which you choose whether usage-based alerts are generated. Select the checkbox to enable quota usage alerts. Clear the checkbox to disable quota usage alerts. If enabled, usage-based alerts (warning level) are generated per the usage threshold setting.

#### Usage Threshold

The percentage of the fileset quota that, when reached, issues an alert. This field is required if the **Enable usage alerts** checkbox is enabled.

---

## Filesets panel

Use this panel to create, attach, detach, delete, and view properties for filesets. You can also create and manage FlashCopy images of filesets.

### Actions

**Create** Takes you to a one-step process to create a fileset.

### Properties

Displays and provides a way to set certain properties for the selected fileset or filesets, such as a percentage-used threshold alert.

**Delete** Deletes the selected fileset from SAN File System.

**Attach** Attaches the fileset by allowing you to specify a path and directory location for it within the global fileset.

### Detach

Verifies the filesets to be detached, and allows you to choose a detachment method (normal or forced).

### Details of the File Placement in Pool

Displays statistics for the storage pools that have received or might receive files from the specified fileset.

### Manage FlashCopy Images

Allows you to view and perform actions on the FlashCopy images of the selected fileset.

### Create FlashCopy Images

Creates a FlashCopy image of the selected fileset or filesets.

### Fields

You can select a fileset for which to view additional details or on which to perform one of the actions described above. In addition to its description, the fileset table provides the following information about each fileset:

**Name** The fileset name. When you click a fileset name in the table, you link to additional details about the associated fileset.

**State** The fileset state. The fileset states can be one of the following:

- Attached
- Detached

**Server** The metadata server from which the fileset is currently being served.

### Assigned Server

The metadata server that is manually assigned to serve the fileset.

### Quota (MB)

The quota for the fileset in megabytes.

### Quota Used (%)

The system percentage used by the fileset.

### Allocated (MB)

The number of megabytes used by the fileset.

### Children

The number of subfilesets.

### Most Recent Image

The date and time that the last FlashCopy image was taken for the associated fileset.

---

## FlashCopy Images panel

Use this panel to manage SAN File System FlashCopy images.

You can create, delete, and view details for FlashCopy images, as well as revert a fileset to a FlashCopy image.

### Actions

**Create** Takes you to a wizard for creating a new FlashCopy image.

**Delete** Deletes the selected FlashCopy images.

**Note:** You cannot delete a FlashCopy image that has client activity (that is, session locks open), unless it is a forced delete.

### Revert to

Reverts the associated fileset to the FlashCopy image that you select.

**Note:** If the associated fileset has child filesets, they must first be detached before you can revert the parent fileset to the selected image.

### Details

Displays additional details about the selected FlashCopy image, such as the associated fileset state and the image's directory path.

### Fields

You can select a FlashCopy image on which to view additional details or perform actions. In addition to its description, the table provides the following information about each FlashCopy image:

**Name** Name of the FlashCopy image of the associated fileset.

**State** The state of the FlashCopy image. If the FlashCopy image is in a complete state, it is intact and you can revert the fileset to that FlashCopy image. If the FlashCopy image is in an incomplete state, it contains incomplete files caused by a (forced) volume deletion. You cannot revert a fileset to that FlashCopy image.

**Fileset** Fileset that the image represents.

### Directory Name

Directory name where the image is located.

**Note:** The FlashCopy image directory is the one below the “.flashcopy” directory of the fileset.

### Date and Time

Indicates when the FlashCopy image was taken.

---

## General Properties of Storage Pools panel

This panel enables you to set default storage pool options.

## Fields

You can disable the use of the default storage pool or set a storage pool as the default storage pool.

### Disable

Disables the use of the default storage pool.

### Enable

#### User Storage Pool

Assigns a user storage pool as the default storage pool. Use the drop-down list to view the available storage pools.

---

## Metadata LUNs panel

Use this panel to view detailed information for metadata logical unit numbers (LUNs) that can be seen by SAN File System. The displayed LUNs include read-only, assigned, and available LUNs.

This panel displays metadata LUNs that are read-only, those that have been assigned as SAN File System volumes to the system storage pool, as well as those that are unassigned, but available.

**Tip:** To view only writable metadata LUNs that have not yet been assigned to the system storage pool, select the **Available Metadata LUNs** panel.

## Actions

### Rediscover LUNs

Refreshes the LUN information from the LUNs, not from the cache.

### Details

Displays additional LUN details, such as the OS device path and sector size.

## Fields

The metadata LUN table provides the following information:

### LUN ID

The identifier of the LUN. When you click on a LUN ID in the table, you link to additional details about the associated LUN.

### Volume Name

The volume name, if the LUN is a volume assigned by SAN File System. The system displays a dash (-) if the LUN has not been assigned as a volume.

**State** The state of the LUN. The LUN states can be one of the following:

#### Assigned

Indicates that the LUN is already assigned to the system storage pool as a usable volume.

#### Available

Indicates the usable LUNs that can be seen by SAN File System, but are not currently assigned to a storage pool.

**Error** Indicates that there was an error in determining the properties of the LUN.

**Unknown**

Indicates that the metadata server is not running, so a LUN state could not be determined (that is, the property could not be queried).

**Unusable**

Indicates the LUNs that are unsuitable to be a SAN File System volume. Possible reasons that a LUN might be unusable include:

- The LUN does not support read or write operations.
- There is inconsistent availability of a LUN from all servers in the cluster.
- There are disk or driver configuration errors.
- A LUN type is not recognizable to the system.

**Note:** The unusable state takes precedence over the assigned or available states.

**Size (MB)**

The total size of the LUN in megabytes.

**Vendor**

The name of the vendor of the storage device associated with the LUN.

**Product**

The product name of the storage device that is associated with the LUN. The system displays the following IBM products in addition to other companies' products:

**2105F20**

ESS F20

**2105800**

ESS 800

**2145** SAN Volume Controller and SAN Integration Server

**3552** FASTT 5000

**1722** FASTT 600

**1742** FASTT 700 and 900

**Storage Device WWNN**

The world-wide node name (WWNN) of the hosting storage device. This information is helpful when you need to ensure that all LUNs in a storage pool are from the same storage device.

---

## Placement of Fileset panel

Use this panel to view information about the placement of the specified fileset.

**Attributes**

The table lists the following information:

**Pool** The name of the storage pool associated with the fileset.

**Fileset Usage**

The indicator of whether the storage pool is in use by the fileset.

**Total Rules That Enable Usage**

The number of policy rules that associate the storage pool with the fileset.



---

## Policies panel

Use this panel to perform actions related to SAN File System policies, or to view existing policy properties. You can create, edit, delete, activate and deactivate policies, or edit policy rules.

### Actions

#### Create

Takes you to a wizard for creating a new policy.

#### Properties

Displays additional details about the selected policy and allows you to edit policy rules.

**Note:** Only inactive policies can be changed. Rules for active policies will be listed as read-only.

#### Delete

Deletes the selected, inactive policy or policies from the system.

**Note:** No policy can be deleted when it is in an active state. You must first activate an inactive policy. This action will automatically make the formerly active policy inactive, allowing you to delete it at that point.

#### Activate

Activates the selected, inactive policy while deactivating the currently active one.

### Fields

You can select a policy for which to view additional details or perform actions. The policy table lists the name and description of the policy and also provides the following:

**State** The state of the policy can be **active** or **inactive**.

**Note:** Only one policy can be active at any given time.

#### Last Active

Date and time that the policy was deactivated.

#### Note:

1. All date and time attributes are based in coordinated universal time (UTC).
2. The "Last Active" state will be shown as a dash (-) if the associated policy was never active or is currently active.

#### Modified

Date and time that the policy was either last modified or initially created.

**Note:** All date and time attributes are based in coordinated universal time (UTC).

---

## Policy Properties – Details panel

Use this panel to view a policy's name, description, state (active or inactive), and the last time it was deactivated or modified.

### Attributes

**State** Can be **active** or **inactive**.

**Note:** Only one policy can be active at any given time.

### Last Active

Date and time that the policy was deactivated.

#### Note:

1. All date and time attributes are based in coordinated universal time (UTC).
2. The "Last Active" state will be shown as a dash (-) if the associated policy was never active or is currently active.

### Modified

Date and time that the policy was either last modified or initially created.

**Note:** All date and time attributes are based in coordinated universal time (UTC).

---

## Policy Properties – Rules panel

Use this panel to view policy rules and edit rules for inactive policies.

**Note:** The following information is important to understand when editing rules:

1. Rules cannot be changed while a policy is in the active state.
2. All date and time attributes in these rules are based in coordinated universal time (UTC). Therefore, any time and date entry is assumed to be in universal time. Also, functions (for example, "dayofweek") are based on UTC.

### Attributes

**Rules** You can only edit rules for inactive policies.

**Note:** Although the rules for the active policy will be read-only, you can copy those rules to the clipboard to paste into another policy.

---

## Policy Statistics – Per Pool panel

Use this panel to view information about files placed in storage pools based on policies.

### Attributes

#### Storage Pool Name

The name of the storage pool.

#### Files Placed

The number of times a file was placed in this storage pool.

#### Last File Placed

The last time a file was placed in this storage pool.

---

## Policy Statistics – Per Rule panel

Use this panel to view information about user-defined rules, as well as the default rule.

### Attributes

#### Rule Name

The name of the rule.

#### Position

The ordinal position of the rule. The order is not viewable when you edit a policy. The rule is sequenced automatically by the system.

**Tip:** Rules are evaluated in order. If possible, initially sort the table from lowest number to highest number.

#### Evaluation Errors

The number of times that the rule has caused an error while being evaluated. These are not syntax errors.

#### Evaluations Not Applied

The number of times that the rule has been evaluated and its conditions were false; therefore, the rule was not applied.

#### Applied Evaluations

The number of times that the rule has been evaluated and its conditions were true; therefore, the rule was applied.

#### Last Applied

The last time a rule was applied. This is the last time the rule placed a file into a storage pool.

---

## Privileged Clients panel

Use this panel to view, revoke, or grant root privileges for SAN File System clients.

There two client privilege levels, also called access levels:

**Root** Privileged client access level.

#### Standard

Normal client access level. This access level is the default.

### Actions

#### Revoke Clients Root Privileges

Removes unrestricted access from the selected clients.

**Note:** You can also perform this action from the **Client Sessions** panel.

### Fields

You can select a client for which to revoke root privileges. The Client with Root Privileges table lists the following information:

#### Client Name

The name of the client. This name is provided on the client side.

**Note:** This is not the IP host name.

**Client IP**

The IP address of the client.

**Actions****Grant Root Privilege****Client Name**

Specifies the client name for which to grant unrestricted access.

**Note:** You can also perform this action from the **Client Sessions** panel.

**Add pushbutton**

Adds the client to the list of those with root privileges.

---

## Processes panel

Use this panel to monitor SAN File System processes, also known as “threads.”

**Actions****General Properties**

Displays information related to all the process limits at both cluster and metadata server levels.

**Fields**

The table provides the following information about each process:

**ID** System-assigned identification number for the associated process.

**Operation**

Name of the process.

**Started**

The time the process command was started.

**Running Time**

The total elapsed time that the process has been running.

---

## Processes General Properties – Cluster-Level Details panel

Use this panel to view process limits for the entire cluster.

**Attributes**

The Cluster-Level Process Limits table displays the following information:

**Server Workload**

A per-metadata server value that applies only to workload operations.

**Admin**

A per-metadata server value that applies only to how many admin commands can be running simultaneously from both the SAN File System console and the command-line interface. The Admin Process Limit is primarily relevant to just the master metadata server, and is totally independent of the Server Workload Process Limit.

**Note:** Very few commands, such as ones relating to changing the state of the cluster, can run on a temporary process even if the limit is currently maxed out.

---

## Processes General Properties – Server-Level Details panel

Use this panel to view process limits for the entire cluster.

### Attributes

The Server-Level Process Limits table displays the following information:

#### Server

The name of the associated metadata server.

**Limit** The maximum number of processes (or threads) to be used for the garbage collection of deleted files. The range is from 1 to 4.

**Restriction:** This attribute cannot be reset after the initial installation and is not configurable from either the SAN File System console or the command-line interface.

---

## Remove Volumes panel

Use this panel to unassign volumes from a particular storage pool. You can also select a method of removal (regular or forced).

When you unassign a volume, it is redefined as an available LUN.

Before removing the volume, you must select a client that can see the volume from the **Client** drop-down box.

### Fields

The Volumes Selected for Removal table lists the name and description for each volume, as well as the following information:

**State** The state of the volume. One of three volume states are possible:

**Active** Only active volumes can have data saved to them.

#### Suspended

Volumes cannot have data stored on them when in the suspended state.

#### Being Removed

Volumes cannot have data stored on them when they are being removed.

**Pool** The name of the storage pool to which the volume belongs.

**Exception:** If you entered this panel from the **Storage Pool Properties – Current Volumes** panel, you do not see the Pool field displayed, it is listed in the panel header instead.

#### Size (MB)

The total space available in the associated volume, in megabytes.

#### Used (MB)

The amount of space currently used by the volume, in megabytes.

### Used (%)

The percentage of the total volume size that is currently being used.

---

## Restart Service panel

Use this panel to start or stop the metadata server restart service, or to view information about individual servers.

This service enables probes to periodically check the metadata server condition. The probes automatically restart the metadata servers when necessary, because they continually monitor metadata server state.

**Restriction:** If a metadata server has the restart service enabled, and the metadata server is later manually shut down, the restart service is disabled and you must start it again.

### Actions

#### Enable Service

Enables the metadata server restart service.

#### Disable Service

Disables the metadata server restart service.

### Statistics

Displays additional properties for the selected metadata server, including tuning details, test details and a probe overview.

### Fields

You can select a metadata server in the Server table on which to perform one of the actions described above. The Servers table also provides the following information:

**Server** Lists all of the metadata servers in the SAN File System cluster. Each metadata server name links to statistical details for that particular metadata server in the **Probe Overview** panel.

### Service State

Specifies the state of the automatic metadata server restart service. The state can be one of the following:

#### "Disabled"

The restart service is off.

#### "Enabled"

#### "Running"

The restart service is running.

#### "Standby"

The metadata server has been manually shutdown. The restart service automatically turns itself on when the metadata server is restarted.

**Restriction:** If a metadata server has the restart service enabled, and the metadata server is later manually shut down, the restart service is disabled and you must start it again.

**“Failed”**

The restart service has detected a non-live metadata server too many times in succession.

**“Unknown”**

The state of the restart service is indeterminate because the metadata server could not be reached.

**Last Probe State**

The probed metadata server status. The state can be one of the following:

**“Not Probed”**

The probe has not started because the metadata server restart service is disabled, in standby, or in an aborted state.

**“Probing”**

The metadata server restart service has started the probe.

**“Live Server”**

The metadata server restart service found the metadata server and it is live. There is no need for the metadata server restart service to restart the metadata server.

**“Absent Server”**

The metadata server restart service positively detected that the metadata server is absent and attempts to restart the metadata server.

**“Unknown”**

The liveness and absence tests failed. The metadata server restart service does not attempt to start the metadata server.

**Last Probe**

The date and time of the last probe.

**Probes**

The total number of probes performed thus far.

**Highest Retries**

The highest number of probe retries thus far.

---

## Restart Service Details panel

Use this panel to view information about metadata server restart service probe intervals, liveness test timeouts, and probe retry limits. The restart service probes monitor metadata servers and automatically restart them as needed.

**Attributes**

**Probe Interval (seconds)**

The metadata server restart service starts a probe at this interval.

**Liveness Test Timeout Limit (seconds)**

The maximum time to wait for a metadata server to respond before declaring it dead.

**Liveness Test Try Limit**

The maximum number of times to try to detect the liveness of a metadata server if the metadata server is not declared dead. When this limit is reached, the metadata server restart service is turned off.

---

## Restart Service - Tuning panel

Use this panel to set information about metadata server restart service probe intervals, liveness test timeouts, and probe retry limits. The restart service probes monitor metadata servers and automatically restart them as needed.

### Attributes

#### Probe Interval

The metadata server restart service starts a probe at this interval.

#### Liveness Test Timeout Limit

The maximum time to wait for a metadata server to respond before declaring it dead.

#### Liveness Test Try Limit

The maximum number of times to try to detect the liveness of a metadata server if the metadata server is not declared dead. When this limit is reached, the metadata server restart service is turned off.

---

## Restart Service Statistics – Probe Overview panel

Use this panel to view information about metadata server restart service probes. These probes monitor metadata servers and automatically restart them as needed.

### Attributes

#### Probe State

The probed metadata server status. The possible values are:

##### “Not Probed”

The probe has not started because the metadata server restart service is either disabled, in standby, or in an aborted state.

##### “Probing”

The metadata server restart service has started the probe.

##### “Live Server”

The metadata server restart service found the metadata server and it is live. There is no need for the metadata server restart service to restart the metadata server.

##### “Absent Server”

The metadata server restart service positively detected that the metadata server is absent and attempts to restart the metadata server.

##### “Unknown”

The tests to determine whether the metadata server is live or absent have failed. The metadata server restart service does not attempt to start the metadata server.

#### Restart Service Started

The date and time that the metadata server restart service was enabled for the associated metadata server.

#### Last Probe

The date and time of the last probe.

#### Current Probe Retries

The number of retries, in succession, that are currently being tried.



**Tip:** This value is normally 0.

**Probes**

The total number of probes performed thus far.

**Probe Retries**

The total number of probe retries.

**Highest Retries**

The highest number of probe retries thus far.

**Lowest Retries**

The lowest number of probe retries thus far.

---

## Restart Service Statistics – Test Details panel

Use this panel to view information about Metadata server restart service “liveness” and “absence” tests. Service probes monitor Metadata servers and automatically restart them as needed.

**Attributes**

The Liveness Test table lists the following information:

**Liveness Test Timeouts**

The total number of times the liveness test has taken longer than the test timeout interval.

**Last Liveness Test (msec)**

The time taken by the last liveness test.

**Highest Liveness Test (msec)**

The highest value for the time taken by the liveness test.

**Lowest Liveness Test (msec)**

The lowest value for the time taken by the liveness test.

**Attributes**

The Absence Test table lists the following information:

**Absence Tests**

Counter of the number of times the absence test was started.

**Last Absence Test (msec)**

Time taken by the last absence test.

**Highest Absence Test (msec)**

Highest value for the time taken by the last absence test.

**Lowest Absence Test (msec)**

Lowest value for the time taken by the last absence test.

---

## Revert Fileset to Image panel

Use this panel to review and verify the image for the fileset to be reverted to, as well as select a reversion method (regular or forced).

You cannot revert a fileset to an image if the fileset has child filesets (that is, subfilesets). You must detach child filesets first.

## Attributes

The Image and Fileset to be Reverted table displays the name and description of the selected FlashCopy image so that you can verify your choice before reverting the associated fileset.

**Attention:** More recent images will be deleted after reverting to an older one.

The table also lists the following detail for the selected image:

**State** The state of the FlashCopy image. If the FlashCopy image is in a complete state, it is intact. If the FlashCopy image is in an incomplete state, it contains incomplete files caused by a (forced) volume deletion.

### Fileset

Fileset that will be reverted to the associated image.

### Fileset State

Two fileset states are possible:

- Attached
- Detached

**Attention:** Though not required, you might want to detach any attached filesets before an image reversion.

### Directory Name

Directory name where the image is located.

**Note:** The FlashCopy image directory is the one below the “.flashcopy” directory of the fileset.

### Directory Path

Path where the FlashCopy image directory is located, using the following naming convention:

- The fileset’s attach point
- “/.flashcopy”
- /Directory name

### Date and Time

Indicates when the FlashCopy image was taken.

---

## Roles panel

Use this panel to view the types and corresponding number of roles configured for SAN File System. Roles refer to the level of access that a user has to SAN File System functionality.

### Fields

**Name** The title of the role within the SAN File System. The role name possibilities are:

- Administrator
- Operator
- Monitor
- Backup

**Note:** User roles are shown in ascending order of access and control.

### Defined Users

The number of users who have been granted that role, and its associated access level, within SAN File System. Users are defined by their highest assigned level of access to the system.

---

## Signon panel

Use this panel to log into the SAN File System console.

The rules associated with signing onto the SAN File System console are determined by the Lightweight Directory Access Protocol (LDAP) server in your environment, which manages SAN File System authentication. Please see your LDAP administrator for details regarding your SAN File System user name and password.

### Actions

In addition to LDAP standards, the following login restrictions apply:

#### User name

Must be alphanumeric and no more than 256 characters.

#### Password

Any valid ASCII is accepted up to 256 characters.

---

## Security Log panel

This panel displays administrator login activity.

### Fields

Using the following panel fields, you can set certain parameters to specify which messages the security log retrieves and refresh the log based on those settings.

#### Order and date of log messages

Specifies the way in which you can limit which messages are retrieved. There are two ways in which you can specify which messages to retrieve, based upon when the messages were issued:

##### Most recent first

Retrieves the most recent rows in the log.

##### Most recent last, beginning on this date:

Retrieves rows in the log beginning with the date specified.

**Levels** Filters the messages that are displayed, based on message severity level.

#### Refresh

Reloads the log based on the adjacent log-retrieval settings.

#### Older Messages

Selects rows in the log, downward, relative to its current view.

#### Newer Messages

Selects rows in the log, upward, relative to its current view.

### Actions

**Help** Displays explanation and action information for a selected message.

**Clear Log**

Deletes all messages in the log.

**Fields**

You can select a log entry on which to perform one of the actions described above. The Security Log table fields provide the following details:

**Message ID**

Each message follows this convention:

"XXX" Component

"YY" Subcomponent

"nnnn"  
Message code

"Z" Severity (Severe, Error, Warning, or Information level)

**Date and Time**

The system (not local) time that the message was issued.

**Level** The severity level of the message. Four levels exist:

-  Severe
-  Error
-  Warning
-  Information

**Server** The metadata server name that is the source of the message.

**Message**

The text of the message.

---

## Server Properties – Details panel

Use this panel to view attributes for a selected metadata server.

**Attributes**

In the General Settings table, you can view the metadata server name as well as the following metadata server details:

**State** The state of the metadata server. The metadata server can be in one of the following states:

- Not running
- Failed initialization
- Initializing
- Not added
- Joining
- Offline
- Fully quiesced
- Partly quiesced
- Online
- Unknown

**Last State Change**

The time when the state last changed.

**Target State**

The state the metadata server is heading toward.

**Last Target State Change**

The time when the target state last changed.

**Server Role**

The status of the master-subordinate relationship among the servers.

**Restriction:** Only one metadata server can be the master. The master role is specified during the initial installation.

**Filesets**

The number of filesets being served by the associated metadata server.

**Last Boot**

The date and time of the last boot for the associated metadata server.

**Current Time**

The current system (not local) time.

**Software Version**

The current software version on the cluster. It might not have yet been committed.

In the Network Settings table, you can see the reserve disk space for logs, server-to-server and client-server transport protocols, as well as the IP address. You can also view the following assigned ports:

**Client-Server Port**

This port is used for communication between a metadata server and a SAN File System client using the SAN File System protocol.

**Tip:** The Client-Server port needs to be the same for the client and the server.

**Cluster Management Port**

This port is used for internal communication among the metadata servers within a cluster.

**Heartbeat Port**

This port is used for internal health monitoring of metadata servers within a cluster.

**Administration Port**

This port is used for receiving administrative requests from an administrative client (this applies to both the SAN File System console and the command-line interface).

---

## Servers panel

Use this panel to perform one of the actions described below on SAN File System servers, or to view metadata server properties.

## Actions

### Properties

Displays additional detailed properties of the selected metadata server, including networking properties.

**Start** Attempts to put the metadata server or servers selected in an available clustered state, if they are not already in an available state.

**Note:** Do not use the start action to recycle a metadata server. The selected servers should be started only when they are in one of the following states:

- Not running
- Unknown

### Restrictions:

1. If the master metadata server is “not running” or “unknown,” you cannot start subordinate servers, without also starting the master.
2. The system attempts to match the servers to the cluster’s current target state (online, offline, fully or partly quiesced).

**Stop** Allows you to gracefully stop the selected metadata server or servers. The servers and their associated states, roles, and filesets are displayed for review and confirmation before stop processing is completed.

**Restriction:** Only active servers can be stopped. Active servers are those in one of the following states:

- Offline
- Fully quiesced
- Partly quiesced
- Online

### Add a Server to Cluster

Enables you to add a metadata server to the cluster. The metadata server that you add must be in a “not added” state and does not appear in the selection list. Being in a “not added” state means that the metadata server is available but unclustered.

### Remove from Cluster

Enables you to remove one or more selected metadata servers from the cluster.

## Fields

You can select a metadata server on which to view additional details or perform one of the actions described above. The Servers table lists the metadata server’s state, and the date and time it was last booted. The table also provides the following information:

**Name** The name of the metadata server. Clicking a server name in the table links to additional details about the selected metadata server.

### Server Role

The status of the master-subordinate relationship among the servers.

**Restriction:** Only one metadata server can be the master. The master role is specified during the initial installation.

#### **Filesets**

The number of filesets being served by the associated metadata server.

---

## **SNMP Properties – SNMP Events panel**

Use this panel to specify the severity levels for SNMP traps that you want to be sent to the appropriate SNMP manager.

#### **Attributes**

##### **Send severe events**

Select this checkbox to send SNMP traps for severe events.

##### **Send error events**

Select this checkbox to send SNMP traps for error events.

##### **Send warning events**

Select this checkbox to send SNMP traps for warning events.

##### **Send information events**

Select this checkbox to send SNMP traps for information events.

---

## **SNMP Properties – SNMP Managers panel**

Use this panel to specify up to two destination SNMP managers. SNMP trap messages are sent to SNMP managers.

#### **Attributes**

Complete the fields to specify up to two destination SNMP managers. To enable sending SNMP trap messages to an SNMP manager select the associated checkbox and enter its Destination IP and Destination Port. To disable sending SNMP trap messages, clear the associated checkbox.

**Note:** One of the SNMP managers should be configured to be the Master console to use the Service Alert or remote access features.

#### **SNMP Manager check boxes**

To enable sending SNMP trap messages to an SNMP manager, select a checkbox and complete the corresponding details.

**Note:** When this checkbox is selected, the Destination IP and Port become required.

#### **Destination IP**

(Required if the corresponding SNMP Manager check box is selected.)

Enter an IP address to enable SNMP trap messages to be sent to the associated SNMP manager. You must enter the address in dotted decimal notation.

#### **Destination Port**

(Required if the corresponding SNMP Manager check box is selected.)

Enter an associated port for the intended SNMP manager.

**Note:** The destination port can range between 1 and 65 535.

### SNMP Version

Select one of the following options:

- V1 (default)
- V2C

**Note:** The “C” stands for “Community.”

### SNMP Community

(Optional) Used for authenticating V1 and V2C. (default: public)

---

## Statistics – Client Sessions panel

Use this panel to view lock and lease information about SAN File System client interactions.

### Fields

#### Attribute

Details information for expired and current clients and total client sessions, as well as session, data and byte-range locks.

**Total** The total for all client sessions.

### Attributes

#### Expired Clients

The clients that have expired client-server leases.

**Attention:** An expired lease might indicate the need for action on the client side by the application administrator. When client leases expire, they could lose access to a set of data if another client creates a lock to it.

#### Current Clients

The clients that have valid client-server leases.

#### Total Client Sessions

Total sessions (current or expired) for all clients.

#### Session Locks

The current number of session locks for all clients. Clients must acquire session locks to do any operation with a filename.

#### Data Locks

The current number of data locks held for all clients. Clients must hold data locks in order to cache data pages, attributes of files, and to cache read-only attributes and contents of directories and links.

#### Byte Range Locks

The current number of byte range locks held for all clients. These locks are used to implement lock system calls and have no direct effect on the contents or attributes of SAN File System objects or other file system opportunities.

---

## Statistics – Cluster panel

Use this panel to view high-level information about the workload on the SAN File System cluster master metadata server.



## Attributes

The Cluster Master Workload on Master Server table lists the following information:

### Update Transactions

The total number of metadata transactions relating to metadata updates for system objects.

### Total Transactions

The total number of metadata transactions relating to metadata activity.

### Dirty Buffers

The current number of dirty buffers for system metadata activity. Dirty buffers contain data awaiting input/output (I/O) to disk.

### Clean Buffers

The current number of clean buffers for system metadata activity. Clean buffers contain data, but are available for reuse.

### Free Buffers

The current number of free buffers for system metadata activity. Free buffers are those that are currently not in use.

### Total Buffers

The current number of total buffers for system metadata activity. Total buffers are composed of clean buffers, dirty buffers, and free buffers.

---

## Statistics – Create Report panel

Use this panel to extract high-level overview information about SAN File System components, including statistics on the cluster, servers, metadata server transactions, buffers, and locks, client sessions, filesets, storage pools, volumes, and engines.

### Fields

#### Cluster - Master Workload

Returns high-level information about the workload on the SAN File System cluster master metadata server.

#### Servers

Returns high-level information about the active or inactive status of servers that are part of the SAN File System cluster.

#### Server Transactions - Client Workload

Returns high-level information about client-level workload, read, and read/write transactions being performed on all the servers, including the master metadata server.

#### Server Buffers - Client Workload

Returns high-level information about client-level work regarding dirty, clean, and free buffers being done on all the servers, including the master metadata server.

#### Server Locks - Client Workload

Returns high-level information about client-level work regarding session, data and byte-range locks being done on all the servers, including the master metadata server.

**Client Sessions**

Returns high-level information about lock and lease interactions between SAN File System and its clients.

**Filesets**

Returns high-level information about SAN File System filesets, including attachment status, soft quota and usage thresholds, FlashCopy image details, and metadata transactions.

**Storage Pools**

Returns high-level information about SAN File System storage pools, including size and percentage used.

**Volumes**

Returns high-level information about SAN File System volumes, including activity status and percentage used.

**Engines**

Returns high-level information about SAN File System engines, including warnings, number of engines, as well as memory and CPU usage.

**Fields****Pushbuttons****Create Report**

Produces a statistics report based on your SAN File System component selections for this panel.

---

## Statistics – Engines panel

Use this panel to view high-level information about SAN File System engines, including warnings, number of engines, as well as memory and CPU usage.

**Fields****Attribute**

Detailed information for engines including those with warnings or errors; normal engines; total engines; the memory used and memory available; and the highest CPU used.

**Total** The total for all engines.

**Attributes****Engines with Warnings or Errors**

An engine is considered to have a warning or an error if it has one or more warning or fan states from the following components:

**Temperature**

One or more “Warning” or higher-level thresholds are surpassed.

**Voltage**

One or more “Warning” or higher-level thresholds are surpassed.

**Fans** One or more fans is below the threshold (15%) fan speed.

**Normal Engines**

Engines that are functioning in a “normal capacity,” meaning those engines for which none of the thresholds for temperatures, voltages, or fans are surpassed.

**Attention:** The normal setting will be the default if no temperature or voltage thresholds have been set.

**Total Engines**

All engines, whether they have warnings or errors, or are functioning normally.

---

## Statistics – Filesets panel

Use this panel to view high-level information about filesets, including attachment status, soft quota and usage thresholds, FlashCopy image details and transaction statistics.

**Fields**

**Attribute**

Specifies detailed information for detached, attached and total filesets; soft quota thresholds; FlashCopy images; and transactions. A *transaction* is a work request that is handled by the metadata server. Use the number of transactions performed by each fileset on the metadata server to aid in balancing the workload among all of the metadata servers in the cluster.

**Value** The value of the attribute.

**Attributes**

**Detached Filesets**

The number of filesets that are detached from the global namespace.

**Attached Filesets**

The number of filesets that are attached to the global namespace.

**Total Filesets**

The total number of filesets.

**Filesets above Soft Quota Threshold**

For soft quotas that are enabled, the number of filesets that are over the threshold (when the usage is greater than or equal to the quota setting).

**Filesets above Usage Threshold**

The number of filesets that have reached their percentage-used alert limit.

**FlashCopy Images**

The total number of images of all filesets.

**Last FlashCopy Image**

The date and time that the last image was taken.

**Started Transactions**

The number of transactions that are in progress for filesets being served by the local metadata server.

**Completed Transactions**

The number of transactions that completed successfully for filesets being served by the local metadata server.

---

## Statistics – Report panel

Use this panel to view online, or print high-level overview information about SAN File System components, including statistics on the cluster, servers, server transactions, buffers, and locks, client sessions, filesets, storage pools, volumes, LUNs, and engines.

For an explanation of the statistics that you selected for the report, refer to the corresponding statistics panel related to the items you selected.

#### Fields

#### Pushbuttons

**Close** Closes the report and returns to the **Create Report** panel.

---

## Statistics – Storage Pools panel

Use this panel to view high-level information about SAN File System storage pools, including size and percentage used.

#### Fields

##### Attribute

Detailed information for storage pools including those above and below usage threshold; total storage pools; total volumes on the storage pools; size of the storage pools in megabytes; the percentage used; and the highest percentage used.

**Total** The total for all storage pools.

#### Attributes

##### Storage Pools above Usage Threshold

The number of storage pools that have reached their percentage-used alert limit.

##### Storage Pools below Usage Threshold

The number of storage pools that have not reached their percentage-used alert limit.

##### Total Storage Pools

Total values for all storage pools.

#### Volumes

The number of volumes on all the storage pools.

#### Size (MB)

The total amount of space (used and unused) in the storage pools.

#### Used (%)

The percentage of storage that is used in the storage pools.

#### Highest “Used (%)”

The highest value of percentage used of all the storage pools (that is, the percentage used for the storage pool that is closest to reaching capacity).

---

## Statistics – Servers panel

Use this panel to view high-level information about SAN File System metadata servers, including client workload details.

#### Fields

The Servers table lists the following information:

##### Attribute

Details information for inactive, active and total metadata servers.

**Total** The total for all metadata servers.

### Attributes

#### Inactive Servers

Metadata servers in one of the following states:

- Not running
- Failed initialization
- Initializing
- Not added
- Joining

#### Active Servers

Metadata servers in one of the following states:

- Offline
- Partly quiescent
- Fully quiescent
- Online

#### Total Servers

All the metadata servers that have been added to the cluster, including those that are in an unclustered meta-state.

### Fields

The Server Transactions — Client Workload table pertains to client-level transaction work being done on all the metadata servers, including the master metadata server. The table lists each metadata server's name and a total for all metadata servers regarding the following details:

#### Update

Update ("write") transactions.

**Total** All metadata server transactions ("read" and "write").

### Fields

The Server Buffers — Client Workload table pertains to client-level work, regarding buffers, being done on all the metadata servers, including the master metadata server. The table lists each metadata server's name and a total for all metadata servers regarding the following details:

**Dirty** The current number of dirty buffers for client/user metadata activity. Dirty buffers contain data awaiting input/output (I/O) to disk.

**Clean** The current number of clean buffers for client/user metadata activity. Clean buffers contain data, but are available for reuse.

**Free** The current number of free buffers for client/user metadata activity. Free buffers are those that are currently not in use.

**Total** All buffer types per metadata server.

### Fields

The Server Locks — Client Workload table pertains to client-level work regarding locks, being done on all the metadata servers, including the master metadata

server. The table lists each metadata server's name and a total workload for all metadata servers regarding the following details:

**Session**

The current number of session locks for each metadata server. A session lock holds a reference on an engine. A client must acquire a session lock to do any operation with a filename.

**Data**

The current number of data locks held for each metadata server. Clients must hold data locks in order to cache data pages, attributes of files, and to cache read-only attributes and contents of directories and links.

**Byte Range**

The current number of byte range locks held for each metadata server. These locks are used to implement lock system calls and have no direct effect on the contents or attributes of SAN File System objects or other file system operations.

**Total**

All lock types per metadata server.

---

## Statistics – Volumes panel

Use this panel to view high-level information about SAN File System volumes, including activity status and percentage used.

**Fields**

The Volumes in User Pool table provides information for volumes in user storage pools, which contain user data. The Volumes in System Pool table provides information for volumes in the system storage pool. The system storage pool contains the system metadata that is accessible to all metadata servers in the cluster.

**Attribute**

Specifies detailed information for volumes including inactive, active, and total volumes; size of the volumes in megabytes; the percentage used; and the highest percentage used.

**Total**

Specifies the total value for the attribute.

**Attributes**

**Inactive Volumes**

The volumes that are in one of the following states:

- Suspended
- Being removed

**Active Volumes**

Those volumes that are being used (that is, the volumes are not Suspended or Being removed).

**Total Volumes**

The total number of volumes.

**Size (MB)**

The total amount of volume space.

**Used (%)**

The percentage of storage that is used in the volumes.

**Highest “Used (%)”**

The highest percentage used for the volume that is closest to reaching capacity.

---

## Stop Cluster panel

Use this panel to view and stop all servers in the SAN File System cluster.

**Attributes**

The name is given for each metadata server to be stopped, as well as the following information:

**State** State that the associated metadata server is in, prior to the stop being executed.

**Server Role**

Gives the status of the master-subordinate relationship among the servers.

**Note:** Only one metadata server can be the master. The master role is specified during the initial installation.

**Filesets**

Number of filesets being served by the associated metadata server.

---

## Stop Servers panel

Use this panel to view and stop the selected servers in the SAN File System cluster.

**Attention:** If you select to stop the master metadata server this way, it will disruptively stop all cluster activity and all client sessions.

**Attributes**

The name is given for each metadata server to be stopped, as well as the following information:

**State** State that the associated metadata server is in, prior to the stop being executed.

**Note:** Only active servers can be stopped. Active servers are those in one of the following states:

- Offline
- Fully quiesced
- Partly quiesced
- Online

**Server Role**

Gives the status of the master-subordinate relationship among the servers.

**Note:** Only one metadata server can be the master. The master role is specified during the initial installation.

**Filesets**

Number of filesets being served by the associated metadata server.

---

## Storage Pools panel

Use this panel to create and delete storage pools; as well as select a storage pool to serve as the system default. You can also view storage pool details.

### Actions

**Create** Takes you to a wizard for creating a new storage pool.

### Properties

Displays, and allows you to set, certain properties for the selected storage pool or storage pools, such as current volumes and available LUNs.

**Delete** Deletes the selected storage pool from SAN File System.

**Note:** You can only delete user storage pools. If you attempt to delete the user default or system storage pool, you will get an error message before the final delete action.

### General Properties

Sets default storage pool properties.

**Note:** You cannot make the system storage pool the default storage pool.

### Fields

You can select a storage pool on which to view additional details or perform actions. In addition to its description, the table provides the following information about each storage pool:

**Name** Storage pool name. When you click a storage pool name in this table, you link to additional details about the associated storage pool.

**Type** One of three storage pool types are possible:

#### User storage pools

Data storage pools in which user data gets stored in blocks.

#### User Default storage pool

The user storage pool that data gets assigned to, if not picked up by a policy rule.

#### System storage pool

The storage pool that stores the metadata (regarding filesets, policies, and so forth).

**Note:** It is important to ensure that the system storage pool never runs out of space. You can monitor the system storage pool usage by setting and enabling a usage threshold alert against it on the **Storage Pool Properties – General Settings** panel.

### Size (MB)

Total space available in the associated storage pool.

### Used (MB)

Amount of space currently used by the storage pool.

### Used (%)

Percentage of the total storage pool size that is currently being used.



**Threshold (%)**

Storage pool usage percentage threshold. When the **Used (%)** reaches this threshold, SAN File System will send an alert.

**Volumes**

Number of volumes in the associated storage pool.

---

## Storage Pool Properties – Available Data LUNs panel

Use this panel to select one or more available logical unit numbers (LUNs) to be added as volumes to the associated storage pool.

Available LUNs are both unassigned and writable.

**Fields**

Specify the settings from which to create the Available LUNs table, and press Refresh.

**Client** Selects a client that can see the available LUNs.

**Method of fetching the LUN information**

Selects the method to use in order to fetch the LUN information from cache or rediscover the LUNs.

**Refresh**

Refreshes the Available LUNs table based on the selected client and fetch method.

**Actions****Add as Volumes**

Initiates the **Add Volumes wizard** displaying the Available LUNs that you select from this panel. You can then verify and modify additional settings in the wizard.

**Fields**

The Available LUNs table allows you to select one or more LUNs to add as volumes to the associated SAN File System storage pool. In addition, the table displays the following information:

**LUN ID**

The identifier of the LUN. When you click a LUN ID in this table, you link to additional details about the associated LUN.

**Size (MB)**

The total size of the LUN in megabytes.

**Vendor**

The name of the vendor of the storage device associated with the LUN.

**Product**

The product name of the storage device that is associated with the LUN.

**OS Device Path**

The path that points to the LUN. The system displays a dash (-) if the version cannot be determined.

---

## Storage Pool Properties – Available Metadata LUNs panel

Use this panel to select one or more available logical unit numbers (LUNs) to be added as volumes to the system storage pool.

Available LUNs are both unassigned and writable.

### Actions

#### Add as Volumes

Initiates the **Add Volumes wizard** displaying the available LUNs that you select from this panel. You can then verify and modify additional settings in the wizard.

#### Rediscover LUNs

Refreshes the LUN information from the LUNs, not from the cache.

### Fields

The Available LUNs table allows you to select one or more LUNs to add as volumes to the system storage pool. In addition, the table displays the following information:

#### LUN ID

The identifier of the LUN. When you click on a LUN ID in the table, you link to additional details about the associated LUN.

#### Size (MB)

The total size of the LUN in megabytes.

#### Vendor

The name of the vendor of the storage device that is associated with the LUN.

#### Product

The product name of the storage device that is associated with the LUN.

#### Storage Device WWNN

The world-wide node name (WWNN) of the hosting storage device. This information is helpful when you need to ensure that all LUNs in a storage pool are from the same storage device.

---

## Storage Pool Properties – Current Volumes panel

Use this panel to view and remove volumes associated with the selected storage pool.

### Actions

#### Remove Volumes

Removes from the storage pool any volumes that you select in the table.

### Fields

The Remove Volumes table allows you to select one or more volumes to remove from the associated storage pool. The table displays a description for each volume in addition to the following information:

**Name** Volume name. When you click a volume name in the table, you link to additional details about the associated volume.

- State** Three volume states are possible:
- Active** Only active volumes can have data saved to them.
  - Suspended**  
Volumes cannot have data stored on them when in the suspended state.
  - Being Removed**  
Volumes cannot have data stored on them when they are “Being Removed.”
- Size (MB)**  
Total volume size in megabytes.
- Used (MB)**  
Number of megabytes currently used by the volume.
- Used (%)**  
Percentage of total volume size currently used by the volume.

---

## Storage Pool Properties – Details panel

Use this panel to view additional attributes for a selected storage pool.

### Attributes

You can view the name, partition size, and description for the storage pool. In addition, the Details panel provides the following information:

- Type** Three storage pool types are possible:
- User storage pools**  
Data storage pools in which user data gets stored in blocks.
  - User Default storage pool**  
The user storage pool that data gets assigned to, if not picked up by a policy rule.
  - System storage pool**  
The storage pool that stores the metadata (regarding filesets, policies, and so forth).
- Note:** It is important to ensure that the system storage pool never runs out of space.
- Size (MB)**  
The total space available in the associated storage pool.
- Used (MB)**  
The amount of space being used in the storage pool.
- Used (%)**  
The percentage of the total storage pool size that is currently being used.
- Threshold (%)**  
The storage pool usage percentage threshold. When the **Used (%)** reaches this threshold, SAN File System will send an alert.
- Allocation Size (KB)**  
The size allocated for the storage pool may be one of three possible (kilobyte) values:
- System Automated

- 4 - Fixed
- 128 - Fixed

#### Volumes

Number of volumes in the associated storage pool.

---

## Storage Pool Properties – General Settings panel

Use this panel to change the selected storage pool's name or description, and percentage used threshold alert.

#### Attributes

**Name** Enter a name for the storage pool (required).

**Note:** This field is read-only for the system storage pool, which will always be named "System" storage pool.

#### Description

Enter a description for the storage pool (optional).

#### Enable usage alerts

You can optionally select to enable an alert when the storage pool reaches a particular percentage of system use.

#### Used Threshold

Specify a percentage of system use that, when reached, will cause SAN File System to issue an alert (if the **Enable usage alerts** checkbox is also selected). The range is 1 to 100%.

---

## System Overview panel




The **System Overview** panel is the highest-level monitoring panel in SAN File System. From this panel you can monitor major system components, including cluster state, clients, filesets, servers, hardware, and recent events.

#### Attributes

The Cluster section of the System Overview provides the following details:




##### Cluster state



The states that are associated with various icons, such as these:

-  The cluster state is Not Running, Offline, Fully quiesced, Partly quiesced, or Online.
-  The cluster state is Forming.
-  The cluster state is Unknown.

##### Master Workload Transactions

The current actual per-minute number of workload transactions. The following are the icon indicators, which are trend arrows that indicate graphically the transaction rate changes since the previous update:

-  Major increase
-  Minor increase
-  Stable

-  Minor decrease
-  Major decrease

### Active Clients

The number of clients that currently have one or more sessions.

**Tip:** Active Clients does not represent the number of total client sessions, which is nearly always the number of active clients multiplied by the number of servers.

### Actions

The Panel control section of the System overview allows you to control the panel refresh interval.

### Refresh Interval

Disables the refresh interval or sets the interval to a value of one, two, five or fifteen minutes. The refresh occurs when the items monitored by the **System Overview** panel are active.

### Submit

Submits the refresh interval that was specified.

**Note:** This selection is reflected upon the next panel update, not necessarily when you click **Submit**, unless the panel update interval exceeds the chosen refresh interval.


### Fields

The Activity and Availability per Server section of the System Overview panel lists the following details:

### Aggregate status icon



The icon displayed above the table represents the highest-level status icon that is contained within the table. The icon can be one of the following:









-  Severe
-  Error
-  Warning
-  Unknown
-  Normal

**Server** The list of all servers in the cluster. Each server name links to additional metadata server details. The master metadata server is indicated by the  icon.

### Server State

The state of the associated metadata server can be one of the following:

-  Not running
-  Failed initialization




-  Initializing
-  Not added
-  Joining
-  Offline
-  Fully quiesced
-  Partly quiesced
-  Online
-  Unknown

### Filesets

The number of filesets being handled by the associated metadata server. The number in this field is also a link to additional fileset details. The accompanying bar graphs represent how the fileset load is distributed among the metadata servers. For example, the metadata server with the most filesets, has the longest bar. The bar graphs decrease in length as the number of filesets per metadata server decreases.

### Fileset Alerts

The alert level for the filesets on the associated metadata server. The alert can be one of the following levels:

-  Error — Indicates that there are one or more hard-quota violations.
-  Warning — Indicates that there are not any hard-quota violations, but one or more usage (Used %), or soft quota violations.
-  Normal — Indicates that there are no fileset alerts. All filesets are below quotas and usage (Used %) alerts.

Also listed is the total number of filesets that surpass one of the quota-related thresholds (percentage used, soft quota, or hard quota). The number in this field is also a link to additional fileset details. For each fileset, the statistics are given for any error or warning-level alerts.

### Trans (per min)

The number of fileset transactions, per minute, for the server.

### Load Share (%)






The associated metadata server's percentage share of the total load on all the metadata servers. The accompanying bar graphs represent how the load is distributed among the metadata servers. For example, the metadata server with the largest load share percentage has the longest bar. The bar graphs decrease in length as the load share percentage per metadata server decreases.

### Fields

The Recent Messages section of the System Overview panel displays the ten most recent messages that are in the cluster log, and that match the Message Types setting for this section.

### Aggregate message icon

The icon displayed above the table represents the highest-level message icon that is contained within the table. The icon can be one of the following:

-  Severe
-  Error
-  Warning
-  Unknown
-  Normal

### Actions

**Levels** Filters can be set for the following alert levels that appear in the Recent Messages table:

- Severe
- Error and above
- Warning and above (default)
- All levels, including Information

**Types** Sets a filter of All Messages or Events Only.

**Tip:** An *event* is a significant or major message that will often, but not always, trigger an alert.

### Submit

Submits the choice for message type.

**Note:** This selection is reflected upon the next panel update, not when you click **Submit**, unless the automatic refresh interval is now less than the elapsed time since the last update.

### Fields

#### Message ID

Each message follows this convention:

“XXX” Component

“YY” Subcomponent

“nnnn”  
Message code

“Z” Severity (Severe, Error, Warning, or Information level)

**Level** The severity level of the message. Four levels exist:

-  Severe
-  Error
-  Warning
-  Information

<b>Type</b>	The messages type, which is one of the following categories: <ul style="list-style-type: none"> <li>• Normal — Messages that are not events.</li> <li>• Event — A significant or major message that can often, but not always, trigger an alert.</li> <li>• Audit</li> <li>• Trace</li> </ul>
<b>Server</b>	The metadata server name that is the source of the message.
<b>Date and Time</b>	The system (not local) time that the message was issued.
<b>Message</b>	The text of the message.

---

## Users panel

Use this panel to view all SAN File System users who are defined and configured in the system by your LDAP server. SAN File System allows various levels of access depending on the tasks each user needs to perform.

### Actions

#### Timeout All Authorizations

Will require LDAP server reauthentication for all users.

### Fields

**Name** Lists all users defined on the LDAP server as authorized to perform some level of administrative actions on SAN File System.

**Role** The highest level of access that the associated user has within the system.

#### Authorization

Status of authorization for user's current session. Authorization can be one of two states:

##### Current

Always corresponds with an Authorization Timeout of greater than "0 sec."

##### Not Current

Always corresponds with an Authorization Timeout of "0 sec."

#### Authorization Timeout

Short interval of time (measured in seconds) in which users have permission to make requests. During this time the LDAP server will not be contacted for authentication. If authorization is "Not Current" the value will be "0 sec."

---

## Using the Help Assistant

The Help Assistant provides field-level descriptions for the actions that you can perform, and the information you can find, on each panel of the SAN File System console.

Because the panel help is embedded into the overall console, you can obtain context-sensitive information without interrupting your current activity. Each time you select a different panel, the information in the Help Assistant changes accordingly.



Each Help Assistant frame contains three possible sections, depending on the type of panel elements being described. The three sections are:

**Actions**


Provides details about the tasks available from panel drop-down menus.

**Fields**

Provides details about column headings or other field descriptions for which you can make selections or enter configurable information.


**Attributes**

Provides details about read-only panel attributes, such as those shown on a tabular “notebook” panel displaying property groupings for a particular SAN File System object.

Within each section, the name of the action, field, or attribute appears with a triangular “twistie” icon (  ) next to it. Select the twistie to expand or collapse the item’s description. The item names are initially collapsed to facilitate easy scanning of the available terms and to reduce scrolling. You can choose to select only those items for which you would like additional information and prevent having the Help Assistant frame cluttered with details you do not need.





At the bottom of each Help Assistant frame are links to additional SAN File System topics. These links provide information on related concepts or tasks. When you choose a “Related Topics” link, the information for that topic appears within the Help Assistant frame.



You can launch the Help Assistant by selecting the **Toggle Help Assistant** icon (

) from the right side of the task bar. Because the icon is on a toolbar, you can toggle the Help Assistant on and off by clicking the icon. You can also close the Help Assistant by selecting the close button for the frame in which it displays.

There are six tool bar controls associated with the Help Assistant:

*Table 4. Help Assistant controls*

Control	Action
 <b>Back</b> icon	Displays the last topic you viewed in the Help Assistant.
 <b>Forward</b> icon	Displays the topic in the Help Assistant that you viewed before clicking the <b>Back</b> icon.
<b>Search</b> field and <b>Go</b> button	Perform an information search and bring up the search returns in the Information Center.
 <b>Return</b>	Displays panel help in the Help Assistant for the current console panel.
 <b>Print</b> icon	Prints the displayed Help Assistant topic.

You can also view all SAN File System product and help documentation by selecting the question mark icon (  ) located above the work content area, next to the Help Assistant icon (  ). This action will launch the SAN File System Information Center in a separate window.

---

## Volume Properties – Details panel

Use this panel to view additional attributes for a selected volume.

### Attributes

In addition to its name, ID, and description, the table provides the following information about each volume:

**State** Three volume states are possible:

**Active** Only active volumes can have data saved to them.

**Suspended**

Volumes cannot have data stored on them when in the suspended state.

**Being Removed**

Volumes cannot have data stored on them when they are “Being Removed.”

**Storage Pool**

Name of the storage pool to which the volume belongs.

**Size (MB)**

Total space available in the associated volume.

**Used (MB)**

Amount of space currently used by the volume.

**Used (%)**

The percentage of the total volume size that is currently being used.

**OS Device Path**

The path that points to the SAN Adapter driver.

**Note:** This driver is local to the SAN File System metadata server cluster and is not on the LUN storage device.

---

## Volume Properties – General Settings panel

Use this panel to change the selected volume’s name or description.

---

## Volume Properties – Size panel

Use this panel to increase the size of a volume when the logical unit number (LUN) size has increased.

Determine the current LUN size by selecting a client that can see the LUN in order to create an up-to-date Volume and LUN Size table.

### Actions

**Rediscover**

Creates an up-to-date Volume and LUN Size table with current sizes in megabytes (MB).

**Expand Volume**

Increases the volume size to correspond with an increased LUN size.

---

## Volume Remove Progress panel

Use this panel to view the progress of a volume removal when it takes longer than five seconds.

### Fields

#### When started

Date and time when the removal started.

#### When stopped

Date and time when the removal completed.

---

## Volumes panel

Use this panel to perform actions on a SAN File System volume, or view volume properties.

### Actions

**Add** Takes you to a wizard to add a volume to a SAN File System storage pool.

### Properties

Displays additional detailed properties of the selected volume. You can also change general volume settings and increase the volume size if the corresponding LUN size has increased.

### Remove

Removes and unassigns volumes from a storage pool.

**Tip:** After volumes are unassigned, they become available LUNs and can be reassigned whenever needed.

### Clients that can see the volume

Lists the clients that can currently see the selected volume.

### Activate

Activates the selected volume. Once the volume is activated, data can be saved to it.

### Suspend

Suspends the volume. Once the volume is suspended, data cannot be saved to it while in this state.

### Fields

You can select a volume on which to view additional details or perform actions. In addition to its description, the table provides the following information about each volume:

**Name** Name of the volume. Clicking on the name in the table links to additional details about the associated volume.

**State** The state of the volume. The state can be one of the following:

**Active** Only active volumes can have data saved to them.

#### **Suspended**

Volumes cannot have data stored on them when in the suspended state.

**Being Removed**

Volumes cannot have data stored on them when they are in this state.

**Storage Pool**

The name of the storage pool to which the volume belongs.

**Size (MB)**

The total space available in the associated volume.

**Used (MB)**

The amount of space currently used by the volume.

**Used (%)**

The percentage of the total volume size that is currently being used.



---

## Appendix H. Prerequisites

There are several hardware and software prerequisites that must be confirmed before installing SAN File System.

### Prerequisites

Before installing SAN File System, there are hardware and software prerequisites that must be provided. Lists of requirements and considerations follow:

- Requirements
  - No software may be installed on the metadata servers or the master console that is not part of the preinstalled SAN File System software except for antivirus packages and Virtual Network Computing (VNC). (There is no antivirus requirement for Metadata servers.
  - The SAN configuration should not have a single point of failure. This means that connectivity should be planned to ensure that the loss of an HBA, switch, GBIC, fibre cable, or storage controller can not cause complete loss of connectivity of SAN File System with the storage subsystem.
  - SAN File System metadata servers must have access to all data and metadata LUNs.
  - Separate fabrics should be used for metadata server connectivity. By creating two distinct, non-overlapping paths from each metadata server to storage, more LUNs are possible. If the paths are connected using the same fabric, intra-fabric path overlap would cause four or more paths to be seen by the metadata server, cutting in half the number of useable LUNs.
  - The SAN File System clients must have access to all data LUNs and must be prevented from having access to metadata LUNs.
  - Data LUNs must be able to be shared among hosts that are based on different operating systems.
  - SAN File System metadata server traffic between the SAN network and the storage array must be segregated from the SAN File System client traffic traveling that same path.
  - HBAs for both clients and servers must be isolated from each other to avoid problems associated with them logging in as both hosts and targets.
  - SAN File System fabrics should be isolated from non-SAN File System fabrics on which administrative activities could occur. Administrative activities on switches can affect all users of a switch, even if proper hard zones are present.
- Considerations
  - Client software requirements
  - Number of clients
  - Number of engines per file system
  - Number of file systems per client
  - Size and number of LUNs
  - LDAP server requirements
  - Network requirements, type and connectivity
  - SAN attachment requirements
  - NFS requirements

- Supported disk subsystems
- Supported client hardware
- Application requirements
- Maximum file size
- Maximum number of files
- Additional recommendations
  - Client-cluster and intra-cluster communication traffic are expected to be on the customer LAN.
  - All nodes must be on the same physical network.
  - All hardware engines are expected to be on the same physical network.
  - If multiple subnets are configured on the physical network, it is recommended that all nodes be on the same subnet.
  - The Remote Supervisory Adapter (RSA II) card network port is expected to be on the same physical network.
  - It is recommended that routers and gateways not exist between the clients and the SAN File System server.

---

## Supported browsers

This topic discusses the Web browsers which SAN File System supports.

### Web browser support

You access the SAN File System console and the online documentation through a standard Web browser. SAN File System supports the following Web browsers (others may work, but have not been tested):

- Internet Explorer 6.0 with Service Pack 1 or higher
- Netscape 6.2 or higher

**Note:** Although Netscape 6.2 is supported, Netscape 7.0 or higher is preferred.

### Limitations

The **Back**, **Forward**, **Refresh** or **Reload** functions of either browser are not supported and may cause unexpected rendering problems. Additionally, opening a hyperlink into a separate browser window is not supported.

---

## **Appendix I. Purchasing additional services**

During and after the warranty period, you can purchase additional services, such as support for other IBM and non-IBM hardware, operating systems, and application programs; network setup and configuration; extended hardware repair services; and custom installations. Service availability and name might vary by country.





---

## Appendix J. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
MW9A/050  
5600 Cottle Road  
San Jose, CA 95193  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Trademarks

The following terms are trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the United States or other countries or both:

AIX	AIX 5L	DB2
Enterprise Storage Server	eServer	FlashCopy
HACMP	IBM	IBM logo

Storage Tank  
WebSphere

Tivoli  
xSeries

TotalStorage

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, and Windows NT are trademarks or registered trademarks of Microsoft Corporation.

Red Hat and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc., in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks

of others.  **JAVA**  
COMPATIBLE



---

## Appendix K. Glossary

This glossary includes terms and definitions from:

- *The American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies can be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions are identified by the symbol (A) after the definition.
- The *ANSI/EIA Standard - 440A: Fiber Optic Terminology*, copyright 1989 by the Electronics Industries Association (EIA). Copies can be purchased from the Electronics Industries Association, 2001 Pennsylvania Avenue N.W., Washington, D.C. 20006. Definitions are identified by the symbol (E) after the definition.
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *Storage Networking Dictionary*, available online at the Storage Networking Industry Association (SNIA) Web site:  
[www.snia.org/education/dictionary/](http://www.snia.org/education/dictionary/)
- The Distributed Management Task Force ([www.dmtf.org](http://www.dmtf.org)), copyright 2003 by the Distributed Management Task Force, Inc., 225 SE Main Street Portland, OR 97214. Definitions derived from this book have the symbol (D) after the definition.

This glossary uses the following cross-reference forms:

**See** This refers the reader to one of two kinds of related information:

- A term that is the expanded form of an abbreviation or acronym. This expanded form of the term contains the full definition.
- A synonym or more preferred term

**See also**

This refers the reader to one or more related terms.

**ACLI** See *administrative command-line interface (ACLI)*.

### **administrative command-line interface (ACLI)**

A command-line interface used to administer all aspects of the SAN File System. The ACLI runs on all engines that host metadata servers and the administrative server.

### **administrative log**

A log that maintains a history of messages created by the administrative server.

### **administrative server**

For SAN File System, a set of servlets running within a customized instance of WebSphere® Application Server that handles all SAN File System administrative requests from the SAN File System console. See also *SAN File System console*.

- alert** A message or other indication that identifies a problem or an impending problem.
- audit log**  
A log that maintains the history of all commands issued by any administrator for all metadata servers in the cluster.
- CIM** See *Common Information Model*.
- CIM client application**  
A storage management program that initiates CIM requests to the administrative agent for the device.
- CIM namespace**  
The scope within which a CIM schema applies.
- CIM object manager (CIMOM)**  
The common conceptual framework for data management that receives, validates, and authenticates the CIM requests from the client application and then directs the requests to the appropriate component or device provider.
- CIMOM**  
See *CIM object manager*.
- client** For SAN File System, a client is a system that can access the SAN File System. These clients act as servers to a broader clientele, providing Network File System or Common Internet File System access to the global namespace or hosting applications (such as database servers or Web-hosting services that use multiple servers).
- class** The definition of an object within a specific hierarchy. An object class can have properties and methods and serve as the target of an association.
- CLI** See *Administrative command-line interface*.
- client state manager (CSM)**  
A component of the client kernel that provides protocol support for the client.
- cluster**  
A group of engines that is managed as a set and presents a single point of control for configuration and service activity.
- cluster log**  
A log that maintains a history of messages created by all metadata servers in the cluster.
- cluster state**  
A status condition of the cluster. Cluster states can be inactive (Not running or Forming), active (Online, Offline, Partly quiescent, or Fully quiescent) or unknown. See also *Forming*, *Fully quiescent*, *Not running*, *Offline*, *Online*, and *Partly quiescent*.
- Common Information Model (CIM)**  
A set of standards from the Distributed Management Task Force Inc. (DMTF). CIM provides a conceptual framework for storage management and an open approach to the design and implementation of storage systems, applications, databases, networks, and devices.
- coordinated universal time (UTC)**  
The time scale, based on the System International (SI) second, as defined

and recommended by the Comité Consultatif International de la Radio (CCIR) and maintained (using an atomic clock) by the Bureau International des Poids et Mesures (BIPM).

**CSM** See *client state manager*.

**default user storage pool**

A storage pool that stores file data that SAN File System has not assigned (using the active policy) to a user storage pool, as well as file data that is assigned directly to this storage pool. There is only one default user storage pool; however, you can assign any user storage pool as the default storage pool. See also *user storage pool*

**engine**

The hardware unit that hosts the software for the metadata server.

**event log**

A log that maintains a history of event messages issued by all metadata servers in the cluster.

IBM Term: A log that contains information about events for a particular system or group, for a particular metric, or for all the events that are associated with a specific monitor.

**failover**

The automatic recovery of resources in the event of a network outage, or failure of the hardware or software.

**file-placement rule**

A rule that controls in what pool SAN File System places files in the global namespace. See also *rule* and *global namespace*.

**fileset** A hierarchical grouping of files managed as a unit for balancing workload across a cluster.

**FlashCopy image**

A space-efficient image of the contents of part of the SAN File System at a particular moment.

**Forming**

A status condition where the cluster has a master and is in the process of forming. This state is always the initial one whenever a cluster is newly formed.

**Fully quiescent**

A status condition that cuts off all client communication with the cluster.

**global namespace**

A single file system that provides complete, shared access to both Windows and UNIX clients in the same environment.

**ID** See *identifier*.

**identifier (ID)**

A sequence of bits or characters that identifies a user, program, device, or system to another user, program, device, or system.

**Initializing**

A status condition during which a metadata server or the entire cluster is set up for the first time.

**key** A property that is used to provide a unique identifier for an instance of a class. Key properties are marked with the Key qualifier. (D)



- lease** The amount of time that a client can hold a lock.
- lock** A restriction that allows clients to have exclusive access to files. Types of locks include *data locks*, *session locks*, and *range locks*.
- logical unit (LU)**  
In open systems, a logical disk drive.
- logical unit number (LUN)**  
In the small computer system interface (SCSI) protocol, a unique number used on a SCSI bus to enable it to differentiate between up to sixteen separate devices per SCSI ID address, each of which is a logical unit.
- LU** See *logical unit*.
- LUN** See *logical unit number*.
- managed object format (MOF)**  
A compiled language for defining classes and instances. A MOF compiler offers a textual means of adding data to the CIM Object Manager repository. MOF eliminates the need to write code, thus providing a simple and fast technique for modifying the CIM Object Manager repository. (D)
- master metadata server**  
In SAN File System, the metadata server in a cluster that is responsible for physical-space allocation.
- metadata**  
Data that describes the characteristics of stored data; descriptive data.
- metadata server**  
In SAN File System, a server that offloads the metadata processing from the data-storage environment to improve SAN performance. An instance of the SAN File System runs on each engine, and together the metadata servers form a cluster. See also *cluster*.
- method**  
A way to implement a function on a class.
- MOF** See *managed object format*.
- Not running**  
A status condition where one or more servers in the cluster are not added and therefore the cluster cannot perform any functions.
- OBDC**  
See *one-button data collection*.
- object name**  
An object that consists of a CIM namespace path and a model path. The namespace path provides access to the CIM implementation managed by the CIM agent, and the model path provides navigation within the implementation.
- Offline**  
A status condition during which clients are not being serviced and the cluster is responding only to administrative requests.
- one-button data collection**  
A utility that gathers data for diagnosing errors or failures associated with metadata servers and clients. It is intended primarily for first-failure data-capture capabilities useful for investigating problems upon their initial occurrence, without requiring problem recreation or subsequent tracing.

**Online**

A status condition that indicates the normal operational state for the cluster.

**Partly Quiesced**

A state in which the cluster or server is in a “quiet” client communications mode to allow other operations to occur.

**Partly quiescent**

A status condition that allows existing metadata activity and client communication to continue on the cluster, but prohibits new communication.

**policy** A list of file-placement rules that define characteristics and placement of files. Several policies can be defined within the configuration, but only one policy is active at one time. See also *file-placement rule* and *service-class rule*.

**pool** See *storage pool*.

**property**

An attribute that is used to characterize instances of a class.

**qualifier**

A value that provides additional information about a class, association, indication, method, method parameter, instance, property, or reference.

**quota** A limit on the amount of disk space a user can use.

**root squashing**

The process of a root or Administrator user logging into a client that is not a privileged client.

**rule** The lines within a policy that specify which actions will occur when certain conditions are met. Conditions include attributes about an object (file name, type or extension, dates, owner, and groups) and the fileset name associated with the object.

**SAN File System console**

A Web user interface used to monitor and control the SAN File System remotely by using any standard Web browser.

**schema**

A group of object classes defined for and applicable to a single namespace. Within the CIM agent, the supported schemas are loaded through the managed object format (MOF) compiler.

**security log**

A log that maintains a history of administrative server login activity.

**service location protocol**

A directory service that the CIM client application calls to locate the CIMOM.

**Shutdown**

A status condition that describes when the cluster is shut down as intended.

**SLP** See *service location protocol*.

**Starting**

A status condition when a metadata server is starting as designed but is not ready to accept connections from clients.

**storage pool**

A named set of storage volumes that is the destination for storing client data.

**symbolic link**

A type of file that contains the path name of and acts as a pointer to another file or directory.

**system storage pool**

A storage pool that contains the system metadata (system and file attributes, configuration information, and metadata server state) that is accessible to all metadata servers in the cluster. There is only one system storage pool. See also *metadata server*.

**user storage pool**

An optional storage pool that contains blocks of data that compose the files that are created by SAN File System clients. See also *storage pool* and *default user storage pool*.

**volume**

A labeled logical unit, which can be a physical device or a logical device. For SAN File System, there is a one to one relationship between volumes and LUNs. See also *logical unit number*.

**UTC** See *coordinated universal time*

---

# Index

## A

- about the Administrator's Guide and Reference ix
- access levels 4
- accessibility
  - disability 115
  - keyboard 115
  - shortcut keys 115
- accessing
  - administrative command-line interface 57
  - SAN File System console 57
- activatevol 135
- activating
  - volume 135
- activating a policy 263
- add volumes to a storage pool
  - select client 336
  - select storage pool 336
  - volume settings 337
- adding
  - LUNs to SAN File System 109
  - privileged client 136
  - SNMP manager 137
  - volume to a storage pool 212
- adding a Metadata server to the cluster 137
- adding a volume to 55
- additional services, purchasing 425
- addprivclient 136
- addserver 137
- addsnmpmgr 137
- administrative
  - log 331
  - security 4
  - server 6
- administrative agent
  - starting 307
  - stopping 312
- administrative CLI
  - logging in 129
  - starting a session 118
- administrative CLI (ACLI)
  - accessing 117
- administrative command-line interface
  - accessing 57
- administrative commands 130
- administrative log 40
- administrative server, accessing using a browser 96
- administrative services 44
- administrative user
  - creating 218
- administrative users
  - listing 167, 202
- Administrator user role 5
- AIX client
  - commands 268
  - starting 72
- AIX client,
  - stopping 74

- AIX, client for
  - controlling disk access 271
  - creating 269
  - destroying 269
  - displaying status for 276
  - loading the file-system driver 273
  - mounting the global namespace 275
  - scanning SAN File System for new and removed volumes 271
  - statistics 309
- alerts 6
- antivirus software 14
- API method
  - backing up 65
- assigning
  - filesets to a new metadata server 231
  - filesets to a new metadata server automatically 140
  - volumes to storage pools 51
- attachfileset 139
- attaching a fileset 139
- audit log 40
  - clearing 154
- Audit log 332
- authentication 14
- authorization 5, 14
- authorizations, timing out 109
- autofilesetserver 140
- available data LUN 342
- available metadata LUN 344

## B

- backing up
  - API method 65
  - considerations for 11
  - data 8
  - file-based limitations to 9
  - files 8
  - files using FlashCopy images 34
  - LUN method 65
  - metadata 8
  - system metadata 11
  - using the volume method 12
  - volume-based advantages to 12
  - volume-based limitations to 12
- backup user role 5
- browsers, supported 424
- buildrrscript 141

## C

- cache 17
- capturing first failure data 303
- case sensitivity 21
- catlog 142
- catpolicy 144
- CD, publications x, xi
- chadvclusterconfig 144

- changes
  - summary of xi
- changing metadata server hostnames 98
- chclusterconfig 146
- checking placement of current files 103
- chfileset 148
- chldapconfig 150
- chpool 152
- chvol 153
- CIM (Common Information Model) 25
- clearing
  - audit log 154
  - cluster log 154
- clearlog 154
- client
  - accessing the global namespace 37
  - accessing using a remote console utility 68
  - accessing using SSH 67
  - accessing using telnet 67
  - adding a privileged 136
  - commands 264
  - creating 267, 269, 280, 297
  - data LUN configuration 14
  - debugging for AIX 308
  - debugging for Solaris 308
  - destroying 269, 280
  - displaying status for 276
  - listing active 173
  - loading the file-system driver 267, 273, 278, 279, 297
  - logs and traces 40
  - mounting the global namespace 267, 275, 282, 288, 297
  - privileged 15
  - removing 267, 290
  - removing a privileged 226
  - setting up 267, 297
  - starting AIX 72
  - starting Linux 73
  - starting Solaris 73
  - starting Windows 73
  - stopping AIX 74
  - stopping Linux 74
  - stopping Solaris 75
  - stopping Windows 75
  - unmounting the global namespace 277
- client CLI
  - accessing 117
- client commands 14
- client sessions
  - panel 348
- client, viewing driver version 68
- clients
  - about fileset 31
  - authentication 14
  - authorization 14
  - clustering 14
  - listing access to a volume or LUN 216

- clients (*continued*)
  - native file-system security 19
  - privileged 19
  - restoring 84
  - supported 13
  - UNIX-based 19
- cluster
  - description of 22
  - log 334
  - modifying advanced settings 144
  - modifying settings 146
  - panel 349
  - properties
    - localization panel 354
    - tuning panel 345, 354
  - starting 239
  - stop panel 408
  - stopping 258
  - upgrading 262
  - viewing statistics 243
- cluster configuration, restoring 81
- cluster language 76
- cluster localization 76
- cluster log
  - clearing 154
- cluster logs 42
- cluster tuning 76
- cluster, adding a Metadata server to 137
- cluster, dropping a metadata server from 158
- cluster, starting 77
- cluster, stopping 77
- cluster, upgrading 77
- clustering 14
- collect diagnostic data
  - panel 354
- collectdiag 155
- collecting diagnostic data 155
- command
  - activatevol 135
  - addprivclient 136
  - addserver 137
  - addsnmpmgr 137
  - attachfileset 139
  - autofileservers 140
  - buildddrscript 141
  - catlog 142
  - catpolicy 144
  - chadvclusterconfig 144
  - chclusterconfig 146
  - chfileset 148
  - chldapconfig 150
  - chpool 152
  - chvol 153
  - clearlog 154
  - collectdiag 155
  - detachfileset 156
  - disableConsoleTrace 301
  - disabledefaultpool 157
  - dropserver 158
  - enableConsoleTrace 301
  - exit 125
  - expandvol 159
  - help 126
  - insmod 278, 279
  - labellun 287
  - legacy 160

- command (*continued*)
  - lsadmuser 167
  - lsautorestart 169
  - lsclient 173
  - lsdrfile 176
  - lsengine 178
  - lsfileset 181
  - lsimage 184
  - lslun 187
  - lspolicy 190
  - lspool 192
  - lsproc 195
  - lsserver 197
  - lssnmpmgr 201
  - lstrapsetting 201
  - lsvol 202
  - migratedata 264
  - mkdrfile 207
  - mkfileset 205
  - mkimage 208
  - mkpolicy 209
  - mkpool 211
  - mktruststore 302
  - mkvol 212
  - mount 288
  - obdc 303
  - parameters, standard 120
  - quiescecluster 214
  - quit 127
  - rediscoverluns 215
  - reportclient 216
  - reportfilesetuse 217
  - reportvolfiles 217
  - resetadmuser 218
  - restartengine 219
  - resumecluster 220
  - reverttoimage 220
  - rmdrfile 223
  - rmfileset 222
  - rmimage 223
  - rmpolicy 224
  - rmpool 225
  - rmprivclient 226
  - rmsnmpmgr 227
  - rmstclient 267, 290
  - rmvol 227
  - sanfs\_ctl 291, 293, 294
  - sanfs\_ctl stats 309
  - sanfs\_ctl trace 308
  - sanfsctl 290
  - setautorestart 229
  - setdefaultpool 231
  - setfileservers 231
  - setmaster 232
  - setoutput 127
  - setport 235
  - settrap 236
  - setupsfs 295
  - setupstclient 267, 297
  - SHOW 160
  - startautorestart 238
  - startcluster 239
  - startConsole 308
  - startengine 240
  - startmetadatacheck 240
  - startserver 242
  - statcluster 243

- command (*continued*)
  - statengine 246
  - statfile 249
  - statfileset 252
  - statldap 253
  - statpolicy 254
  - statserver 256
  - stfsclient 269, 280
  - stfsdebug 308
  - stfsdisk 271
  - stfsdriver 273
  - stfsmount 275, 282
  - stfsstat 309
  - stfsstatus 276
  - stfsumount 277
  - stopautorestart 258
  - stopcluster 258
  - stopConsole 313
  - stopengine 259
  - stopmetadatacheck 260
  - stopserver 261
  - suspendvol 262
  - tank extractbootrecord 313
  - tank lscluster 314
  - tank lsdisklabel 317
  - tank lsversion 316
  - tank resetcluster 319
  - tank resetversion 319
  - tankpasswd 129
  - TRACE 160
  - usepolicy 263
  - user assistance for 122
- commands
  - administrative 130
  - AIX client 268
  - client 264
  - common client 264
  - Linux client 278
  - modes 119
  - service 299
  - Solaris client 287
  - Windows client 295
- commands, using metadata server
  - operating system 59
- common client
  - commands 264
- Common Information Model (CIM) 25
- communication between metadata servers 23
- components
  - SAN File System 25
- configuration
  - data LUN 14
- Configuration/Setup Utility 88
- configuring cluster language 76
- configuring cluster tuning 76
- considerations
  - file name 22
  - fileset 32
  - FlashCopy images 36
  - nested fileset 33
- considerations for volumes 55
- container (see fileset) 27
- controlling disk access for a client for AIX 271
- conventions, syntax diagram 123

- copy on write
  - description of 34
- copying a policy 103
- create storage pool
  - add volume 361
  - introduction 362
  - select client 363
  - set properties 362
  - verify settings 363
  - volume settings 364
- creating
  - client 267, 297
  - client for AIX 269
  - client for Linux 280
  - disaster-recovery dump files 207
  - disaster-recovery script files 141
  - filesets 205
  - FlashCopy image 208
  - FlashCopy image details 93
  - FlashCopy images 35
  - policies 209
  - storage pools 211
  - truststore 302
- creating a policy 104

## D

- data cache 17
- data migration
  - definition 27
  - prerequisites 29
- data, collecting diagnostic 155
- date and time formats 325
- debugging client for AIX 308
- debugging client for Solaris 308
- default storage pool
  - disabling 157, 159
- default storage pool, disabling 107
- default storage pool, setting 107
- deleting
  - disaster-recovery dump files 223
  - FlashCopy image 93, 223
  - storage pool 225
- deleting a fileset 91
- deleting a policy 104
- deleting a recovery file 79
- deleting a storage pool 107
- deleting SNMP managers 62
- destroying
  - client for AIX 269
  - client for Linux 280
- detachfileset 156
- detaching a fileset 156
  - fileset, detaching 91
- details
  - active and inactive policies 190
  - administrative users 167
  - available LUNs 187
  - client 173
  - disaster-recovery files 176
  - engines 178
  - fileset 181
  - FlashCopy image 184
- details of client session
  - panel 368
- details panel
  - policy properties 387

- diagnostic data collection
  - engines 86
- direct I/O 17
- disableConsoleTrace 301
- disabledefaultpool 157
- disabling
  - automatic restart of the metadata server 258
  - default storage pool 157, 159
  - SAN File System console Web server 301
- disaster recovery commands 313, 314, 316, 317, 319
- disaster-recovery dump files
  - creating 207
  - deleting 223
- disaster-recovery files
  - listing 176
- disaster-recovery script files
  - creating 141
- discovering LUNs 109
- displaying
  - automatic restart of the metadata server 169
  - log-file content 142
  - long-running processes 195
  - policy rules 144
  - SNMP managers 201
  - statistics for a client for AIX 309
  - statistics for a client for Solaris 309
  - statistics for a client for Windows 309
  - status for a client for AIX 276
- displaying version of client driver 68
- dropping a metadata server from the cluster 158
- dropserver 158
- dynamic fileset 24

## E

- enableConsoleTrace 301
- enabling
  - automatic restart of the metadata server 238
- ending a sfscli session 125, 127
- engine
  - accessing using SSH 85
  - definition 29
  - listing 178
  - powering off 259
  - restarting 87, 219
  - starting 240
  - viewing statistics 246
- engine, powering off 87
- engines
  - collecting diagnostic data on 86
  - listing 87
  - properties
    - fans panel 374
    - power panel 374
    - summary panel 375
    - temperatures panel 376
    - timeouts panel 377
    - vital engine data panel 378
    - voltages panel 378

- event
  - log 333
- event log 42
- exit 125
- expandvol 159

## F

- failover 24
- FIFO objects 20
- file
  - sharing 16
  - viewing statistics 249
- file names
  - considerations 22
- file permissions 15
- file placement, policy-based 48
- file-placement-rule syntax
  - conventions 323
- file-system drive
  - version of 276
- files
  - backing up 8
  - limitations to backing up 9
  - listing, in a volume 217
  - restoring 8
- files, checking placement of current 103
- fileset
  - about clients 31
  - about metadata servers 31
  - about storage pools 32
  - attaching 30, 139
  - considerations 32
  - considerations for nested 33
  - creating 30, 205
  - creating objects in 30
  - description of 30
  - detaching 156
  - listing 181
  - listing usage 217
  - managing storage space in 111
  - modifying settings 148
  - permissions 32
  - placing in storage pools 30
  - properties
    - Details panel 379
    - Quota Options panel 381
  - quotas 33
  - reassigning to a metadata server 231
  - reassigning to a metadata server automatically 140
  - removing 222
  - viewing statistics 252
- fileset (container)
  - panel, attach 340
  - properties
    - General Settings panel 381
- fileset, attaching 90
- fileset, configuring on UNIX-based client 71
- fileset, configuring on Windows client 71
- fileset, deleting 91
- fileset, sharing UNIX 69
- fileset, sharing Windows 70
- filesets
  - panel 382

first failure data capture 252, 303

FlashCopy image

creating 93, 208

deleting 93, 223

listing 93, 184

reverting to 220

FlashCopy image, reverting to

previous 93

FlashCopy image, viewing details 94

FlashCopy images

about backing up files 34

about creating 35

about reverting files 34

considerations 36

description of 34

space for 37

FlashCopy Images of Filesets

panel 357

FlashCopy Images of Filesets panel 357

functions

date and time 325

numerical 323

string 323

## G

getting help 4, 126

global fileset

description of 30

global namespace 37

attaching a fileset to 139

client access to 37

shared access 38

structure of 38

granting user access 108

guidelines, naming 119

## H

hardware

restoring 79

HBA, monitoring metadata server

statistics 59

help 126

general 329

telephone 330

Help Assistant

using 417

help parameters, standard 122

## I

insmod 278, 279

interaction between metadata servers 23

## L

label, volume 54

labellun 287

LDAP

modifying configuration for 150

viewing settings 253

lease 39

legacy 160

legacy CLI commands, running 160

limitations 56

UNIX-based clients 20

Windows-based client 22

limited warranty x

Linux client

commands 278

starting 73

Linux client,

stopping 74

Linux, client for

creating 280

destroying 280

loading the file-system driver 278,

279

mounting the global namespace 282

listing

active and inactive policies 190

active clients 173

administrative users 167, 202

automatic restart of the metadata

server 169

available LUNs 111, 187

disaster-recovery files 176

engines 87, 178

files in a volume 217

files on a volume 111

fileset 181

filesets usage 217

FlashCopy image 93, 184

long-running processes 195

LUNs 111

metadata server 197

policies 104

recovery files 79

SNMP managers 201

storage pool 107, 192

trap settings 201

volumes 111

volumes in a storage pool 107

listing parameters, standard 122

loading the client file-system driver 267,

273, 278, 279, 297

localization panel, cluster 354

lock 39

lock cache 17

log

administrative 40, 331

audit 40

client 40

cluster 42, 334

description of 40

event 42, 333

metadata server 42

security 42, 333

trace 43

log file

displaying the content of 142

logging in to the administrative CLI 129

logging into the SAN File System

console 396

logical unit (LUN) 54

logs 331

clearing 154

long-running processes

listing 195

lost+found directory 18

lsadmuser 167

lsautorestart 169

lsclient 173

lsdrfile 176

lsengine 178

lsfileset 181

lsimage 184

lslun 187

lspolicy 190

lspool 192

lsproc 195

lsserver 197

lssnmpmgr 201

lstrapsetting 201

lsvol 202

LUN

configuration 14

listing client with access to 216

LUN method

backing up 65

LUNs

adding 109

listing 111

listing available 111, 187

rediscovering 215

## M

managing allocated storage space 111

managing free storage space 111

managing storage space 111

master console

accessing remotely 94

restoring 79

Master console 43

master server, changing 97

message IDs 42

metadata

restoring 83

server 44, 45

services 44

starting the checker 240

stopping the checker 260

metadata cache 17

metadata restart service 25

metadata server

disabling automatic restart 258

enabling automatic restart 169, 238

listing 197

restarting 25

setting the master 232

starting 242

starting all in the cluster 239

stopping 261

stopping all in the cluster 258

viewing statistics 256

Metadata server

software version 397

metadata server HBA statistics,

monitoring 59

metadata server logs 42

metadata server operating system

commands, using 59

metadata server restart service 99

metadata server restart service,

stopping 100

metadata server, adding to the

cluster 96



- Metadata server, adding to the
  - cluster 137
- metadata server, dropping from the
  - cluster 158
- metadata server, removing from the
  - cluster 99
- metadata server, starting 99
- metadata server, stopping 100
- metadata servers
  - about fileset 31
  - communication between 23
- metadata volume
  - limitations 56
- migratedata 264
- migrating data to SAN File System 264
- migration, data 27, 29
- mkdrfile 207
- mkfileset 205
- mkimage 208
- mkpolicy 209
- mkpool 211
- mktruststore 302
- mkvol 212
- modes, command 119
- modifying
  - advanced cluster settings 144
  - cluster settings 146
  - fileset settings 148
  - pool settings 152
  - volume settings 153
- modifying LDAP configuration 150
- monitor user role 5
- monitoring metadata server HBA
  - statistics 59
- mount 288
- mounting the global namespace on a
  - client for AIX 275
- mounting the global namespace on a
  - client for Linux 282
- mounting the global namespace on a
  - client for Solaris 288
- mounting the global namespace on the
  - client 267, 297

## N

- named pipes 20
- naming guidelines 119
- navigating by keyboard 115
- non-uniform data LUN configuration 14
- notices 427
- notices used in this guide ix
- NTFS
  - differences 21
  - unsupported features 22

## O

- obdc 303
- operating system
  - restoring 79
- operating system commands, using 59
- operator user role 5
- opportunistic locks (oplocks) 17
- orphaned objects 18

- output format
  - setting 127
- overview
  - SAN File System 1
- ownership, taking AIX filesets 103

## P

- panel
  - available data LUNs 342
  - available metadata LUNs 344
  - cluster 349
  - collect diagnostic data 354
  - policies 386
  - Signon 396
  - system overview 413
- panels
  - volumes 420
- permissions
  - fileset 32
- permissions, file 15
- placement of current files, checking 103
- placement of fileset
  - panel 385
- placement policy rule syntax 323
- policies
  - listing 104
  - listing active and inactive 190
  - removing 224
- policies and rules
  - using 49
- Policies panel 386
- policy
  - activating 263
  - copying 103
  - creating 209
  - properties
    - details panel 387
    - rules panel 387
  - viewing statistics 254
- policy rules 323
  - displaying 144
- Policy wizard
  - Add Rules 359
  - Edit Rules 360
  - High-Level Settings 361
  - Introduction 361
- policy, creating 104
- policy, deleting 104
- pool
  - creating 211
  - modifying settings 152
  - setting the default 231
- port
  - setting values for 229, 235
- powering off an engine 259
- powering off the engine 87
- prerequisites
  - data migration 29
  - general 423
- privileged client
  - adding 136
  - removing 226
- privileged clients 15, 19
  - panel 388
- privileges, setting UNIX
  - administrative 71

- privileges, setting Windows
  - administrative 71
- processes
  - listing long-running 195
- publications x, xi
- publications CD x, xi

## Q

- quiescecluster 214
  - quiescing 214
  - resuming 220
- quiescing the cluster 214
- quit 127
- quotas
  - fileset 33

## R

- reassigning
  - filesets to a new metadata server 231
- rebooting engine 219
- rebooting the engine 87
- recovery file, deleting 79
- recovery files
  - listing 79
- rediscovering LUNs 215
- rediscoverluns 215
- release notes x, xi
- remote access 46
- remote console utility
  - accessing a client using 68
- Remote Supervisor Adapter II,
  - accessing 85
- removing 56
  - client 267, 290
  - fileset 222
  - policies 224
  - privileged client 226
  - SNMP manager 227
  - volume 227
- removing down-level metadata server
  - software 102
- removing volumes from a storage
  - pool 112
- reportclient 216
- reportfilesetuse 217
- reportvolfiles 217
- requirements 423
- resetadmuser 218
- resetting
  - administrative user 218
- restart service 99
- restart service, stopping 100
- restartengine 219
- restarting
  - metadata server 25
  - metadata server automatically 238, 258
- restarting engine 219
- restarting the engine 87
- restoring
  - considerations for 11
  - data 8
  - file-based limitations to 9
  - files 8



- restoring *(continued)*
  - metadata 8
  - system metadata 11
  - using the volume method 12
  - volume-based advantages to 12
  - volume-based limitations to 12
- resumecluster 220
- resuming the cluster 220
- reverting
  - files using FlashCopy images 34
- reverting to a FlashCopy image 93, 220
- reverttoimage 220
- rmdrfile 223
- rmfileset 222
- rmimage 223
- rmpolicy 224
- rmpool 225
- rmprivclient 226
- rmsnmpmgr 227
- rmstclient 267, 290
- rmvol 227
- rogue metadata server 24
- roles 108
- root squashing 15
- RSA II, accessing 85
- rule functions 323
- rules panel, policy properties 387
- running administrative commands 118
- running legacy CLI commands 160

## S

- safety information x, xi
- safety notices, translated x
- SAN connectivity
  - restoring 80
- SAN File System
  - components 25
  - description of 1
  - getting help 4
  - getting started with 3
  - major features 2
  - overview 1
  - software 3
- SAN File System accessibility
  - features 115
- SAN File System console
  - accessing 57
- SAN File System console Web server
  - disabling 301
  - starting 308
  - stopping 301, 313
- sanfs\_ctl 291, 293, 294
- sanfs\_ctl stats 309
- sanfs\_ctl trace 308
- sanfsctl 290
- security
  - administrative 4
  - for native client file systems 19
  - log 333
- security log 42
- server
  - administrative 6
  - metadata 44, 45
  - properties
    - panel 397

- servers
  - panel 399
  - stop panel 408
- service
  - phone 329
- Service alert 46
- services
  - administrative 44
  - metadata 44
  - storage management 45
- setautorestart 229
- setdefaultpool 231
- setfilesetserver 231
- setmaster 232
- setoutput 127
- setport 235
- setting
  - master metadata server 232
  - SNMP traps 236
- setting output format 127
- setting port values 229, 235
- setting the default storage pool 231
- setting up the clients 267, 276, 297
- settings
  - LDAP 253
  - listing, trap 201
- settrap 236
- setupsfs command 295
- setupstclient 267, 297
- sfscli
  - ending the session 125, 127
  - getting help 126
- sharing files 16
- sharing UNIX filesets 69
- sharing Windows filesets 70
- SHOW command 160
- signing onto the SAN File System
  - console 396
- Simple Network Management Protocol (SNMP)
  - components 47
  - traps 6
- skills needed to administer this
  - product ix
- SNMP
  - adding a manager 137
  - listing managers 201
  - removing a manager 227
  - setting a trap 236
- SNMP (Simple Network Management Protocol)
  - components 47
  - traps 6
- SNMP manager, adding 62
- SNMP managers, deleting 62
- SNMP managers, modifying 63
- SNMP managers, viewing 64
- SNMP traps, modifying 63
- SNMP traps, setting up 63
- software
  - restoring 81
- Solaris client
  - commands 287
  - starting 73
  - stopping 75
- Solaris, client for
  - mounting the global namespace 288

- Solaris, client for *(continued)*
  - statistics 309
- spare metadata server 24
- standard
  - command parameters 120
  - help parameters 122
  - listing parameters 122
- startautorestart 238
- startCimom 307
- startcluster 239
- startConsole 308
- startengine 240
- starting
  - administrative CLI session 118
  - AIX client 72
  - all metadata servers in the
    - cluster 239
  - Linux client 73
  - metadata checker 240
  - SAN File System console Web
    - server 308
  - Solaris client 73
  - Windows client 73
- starting an engine 240
- starting an metadata server 242
- starting the administrative agent 307
- startmetadacheck 240
- startserver 242
- statcluster 243
- statengine 246
- statfile 249
- statfileset 252
- static fileset 24
- statistics
  - client for AIX 309
  - client for Solaris 309
  - client for Windows 309
  - client sessions panel 401
  - cluster 243
  - cluster panel 402
  - create report panel 402
  - engine 246
  - engines panel 403
  - file 249
  - fileset 252
  - filesets panel 404
  - metadata server 256
  - policy 254
  - report panel 405
  - servers panel 405
  - storage pools panel 405
  - volumes panel 407
- statistics, monitoring metadata server
  - HBA 59
- statldap 253
- statpolicy 254
- statserver 256
- stfsclient 269, 280
- stfsdebug 308
- stfsdisk 271
- stfsdriver 273
- stfsmount 275, 282
- stfsstat 309
- stfsstatus 276
- stfsumount 277
- stop panel, cluster 408
- stop panel, servers 408

- stopautorestart 258
- stopCimom 312
- stopcluster 258
- stopConsole 313
- stopengine 259
- stopmetadatacheck 260
- stopping
  - AIX client 74
  - all metadata servers in the
    - cluster 258
  - Linux client 74
  - metadata checker 260
  - SAN File System console Web
    - server 301, 313
  - Solaris client 75
- stopping a metadata server 100
- stopping an metadata server 261
- stopping client for Windows
  - Windows client 75
- stopping metadata server restart
  - service 100
- stopping the administrative agent 312
- stopping the cluster 77
- stopping the engine 87
- stopserver 261
- storage management 48
- storage management services 45
- storage pool
  - adding volumes to 212
  - assigning volumes to 51
  - creating 211
  - deleting 225
  - description of 51
  - disabling the default 157, 159
  - listing 107, 192
  - listing volume in 107
  - managing storage space in 111
  - setting the default 231
  - system 52
  - user 52
- storage pool, creating 106
- storage pool, default 107
- storage pool, deleting 107
- storage pools 55
  - about fileset 32
- storage space
  - managing 111
- storage subsystem, supported 50
- summary of changes xi
- support
  - general 329
  - telephone 330
- supported browsers
  - limitations 424
- suspending a volume 262
- suspendvol 262
- syntax conventions, file-placement-rule 323
- syntax diagram conventions 123
- system metadata
  - backing up and restoring 11
- system overview
  - panel 413
- system storage pool
  - description of 52

## T

- taking ownership, filesets 71
- tank extractbootrecord 313
- tank lscluster 314
- tank lsdisklabel 317
- tank lsversion 316
- tank resetcluster 319
- tank resetversion 319
- Tank.Bootstrap file 313, 319
- tankpasswd 129
- telnet, accessing the client using 67
- time formats 325
- TRACE command 160
- trace log 43
- traces
  - client 40
- trademarks 428
- traps
  - listing settings 201
- traps, Simple Network Management Protocol (SNMP) 6
- truststore
  - creating 302
- tuning panel, cluster 345, 354

## U

- unbuffered I/O 17
- uniform data LUN configuration 14
- UNIX-based clients 19
  - limitations of 20
- unmounting the global namespace on a
  - client for AIX 277
- upgrading cluster 262
- upgrading the cluster 77
- usepolicy 263
- user access, granting 108
- user assistance for commands 122
- user authorization 5
- user authorizations, timing out 109
- user data
  - restoring 84
- user interface
  - Web-based 53
- user role
  - Administrator 5
  - backup 5
  - monitor 5
  - operator 5
- user storage pool
  - description of 52
- utility
  - startCimom 307
  - stopCimom 312

## V

- viewing
  - cluster statistics 243
  - engine statistics 246
  - file statistics 249
  - fileset statistics 252
  - LDAP settings 253
  - metadata server statistics 256
  - policy statistics 254
- viewing FlashCopy image details 94

## volume

- activating 55, 135
- adding to a storage pool 212
- adding to storage pools 55
- considerations 55
- description of 54
- limitations in the system storage
  - pool 56
- listing client with access to 216
- listing files in 217
- listing files on 111
- listing in a storage pool 107
- modifying settings 153
- remove progress panel 420
- removing 227
- removing from a storage pool 56
- scanning SAN File System for 271
- suspending 55, 262
- volume from a storage pool 56
- volume label 54
- volume-based backup and restore 12
- volumes
  - advantages to backing up 12
  - assigning to storage pools 51
  - limitations to backing up 12
  - listing 111
- volumes in the system storage pool 56
- Volumes panel 420
- volumes, adding to storage pool 110

## W

- Web browsers supported 424
- Web sites xi, 329
- who should use this guide ix
- Windows client
  - commands 295
  - starting 73
  - stopping 75
- Windows-based client
  - administrative privileges 20
  - case sensitivity 21
  - difference from NTFS 21
  - file management 20
  - limitations of 22
  - NTFS features 20
- Windows, client for
  - statistics 309







Printed in USA

GA27-4317-01

