Data ONTAP™ Release 5.0

System Adminstrator's Guide

Network Appliance, Inc.
2770 San Tomas Expressway
Santa Clara, CA 95051, USA
Tel: +1 408 367-3000
Fax: +1 408 367-3151
Support tel: (888) 4-NETAPP
Support email: support@netapp.com
Information email: info@netapp.com
Web: http://www.netapp.com

Part number 210-01819
March 1998

Copyright

Copyright © 1998 Network Appliance, Inc. All rights reserved. Printed in the U.S.A.

No part of this book covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Portions of this product are derived from the Berkeley Net2 release and the 4.4-Lite-2 release, which are copyrighted and publicly distributed by The Regents of the University of California.

Copyright © 1980–1995 The Regents of the University of California. All rights reserved.

Portions of this product are derived from NetBSD, which is copyrighted and publicly distributed by Carnegie Mellon University.

Copyright © 1994, 1995 Carnegie Mellon University. All rights reserved. Author Chris G. Demetriou.

Permission to use, copy, modify, and distribute this software and its documentation is hereby granted, provided that both the copyright notice and its permission notice appear in all copies of the software, derivative works or modified versions, and any portions thereof, and that both notices appear in supporting documentation.

CARNEGIE MELLON ALLOWS FREE USE OF THIS SOFTWARE IN ITS “AS IS” CONDITION. CARNEGIE MELLON DISCLAIMS ANY LIABILITY OF ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.

Software derived from copyrighted material of The Regents of the University of California, Carnegie Mellon University, and Network Appliance are subject to the following license and disclaimer:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notices, this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

This product includes software developed by the University of California, Berkeley and its contributors.

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Network Appliance reserves the right to change any products described herein at any time, and without notice. Network Appliance assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by Network Appliance. The use and purchase of this product do not convey a license under any patent rights, trademark rights, or any other intellectual property rights of Network Appliance.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademarks

Network Appliance, NetCache, Data ONTAP, Snapshot, WAFL, Web Filer, FilerView, BareMetal, and SecureShare are trademarks of Network Appliance, Inc. FAServer, NetApp, and the Network Appliance logo are registered trademarks of Network Appliance, Inc.

All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.

Communications regulation information

Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the users will be required to correct the interference at their own expense.

Cables

To maintain compliance with applicable emission standards, use cables shielded with metallic RFI/EMI connector hoods on this device.

Canada notice

This digital apparatus does not exceed the Class A limits for radio noise emissions from digital apparatus set out in the radio interference regulations of the Canadian Department of Communications.

Avis Canadien

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe A prescrites dans le Règlement sur le brouillage radioélectrique édicté par le ministère des Communications du Canada.

BZT Hinweis

Hiermit wird bescheinigt, daß dieses Gerät in Übereinstimmung mit den Bestimmungen der BAPT-AmtsblVfg 243/1991 funkentstört ist. Der vorschriftsmäßige Betrieb mancher Geräte (z.B. Meßsender) kann allerdings gewissen Einschränkungen unterliegen. Beachten Sie deshalb die Hinweise in der Bedienungsanleitung.

Dem Bundesamt für Zulassungen in der Telekommunikation wurde das Inverkehrbringen dieses Gerätes angezeigt und die Berechtigung zur Überprüfung der Serie auf die Einhaltung der Bestimmungen eingeräumt.

Safety warnings

Failure to follow the directions contained in the following statements could result in bodily harm or loss of life.

- To reduce the risk of personal injury from hot surfaces, allow the internal components time to cool before touching or installing options.
- To reduce the risk of personal injury or damage to the equipment, ensure that the equipment is properly supported or braced when installing options.
- This equipment is designed for connection to a grounded outlet. The grounding type plug is an important safety feature. To avoid the risk of electrical shock or damage to the equipment, do not disable this feature.
- This equipment is furnished with one or more lithium batteries. There is a danger of explosion and risk of personal injury if the battery is incorrectly replaced or mistreated. Do not attempt to recharge the battery, disassemble it, or dispose of it in a fire.

Table of Contents

Preface xxi

About this guide xxi

If you are installing a filer for the first time xxi

If you are upgrading a filer xxi

Audience xxii

Terminology and conventions xxii

Keyboard conventions xxii

Special messages xxiii

Chapter 1:
Introducing Network Appliance Filers 1

Filer features 1

Multiprotocol support 1

Administration protocols 2

Disk and file system management features 2

What to do next 3

Chapter 2:
Routine Filer Administration 5

Using the administration host 6

The root volume 7

Editing configuration files 7
From an NFS client 8
From a CIFS client 8
Obtaining access to the filer shell 9
Halting and rebooting the filer 11
Understanding the default configuration 13
The root directory 13
The /home directory 15
The /etc/rc file 17
The /etc/rc default contents 17
Editing /etc/rc 20
Changing SNMP commands in /etc/rc 20
Naming conventions for network interfaces 22
Recovering from configuration errors 22
Core files 24
Message logging 25
The facility parameter 25
The level parameter 25
The action parameter 26
Configuring filer options 28
The options command 28
The vol options command 29
Detailed option information 29
Option values 30
Autosupport options 31
CIFS options 32
DNS options 35

HTTP options 35
NFS and (PC)NFS options 36
NIS options 38
RAID options 39
Volume options 39
Miscellaneous options 41
Automatic email to Network Appliance 43
How the autosupport daemon works 43
Contents of automatic email messages 44
Using the options command to configure autosupport 45
Synchronizing filer system time 48
Setting the date and time 48
Synchronizing the filer 48
Maintaining filer security 50
Software licenses 51
Enabling NFS, CIFS, or HTTP service 51
Displaying current license codes 53
Disabling a license 53
Replacing license codes 53
Chapter 3: Managing Disks and File Systems 55
Filer fundamentals 55
Disk concepts 56
Understanding RAID groups 56
Understanding hot spare disks 60
Understanding hot swap 60
Using disks of various sizes 61

Usable space on each disk 62
Handling disk failures 63
Volume concepts 66
Understanding volumes 66
Determining the number of volumes to use 67
Planning a multiple volume configuration 69
About foreign volumes 69
Handling volume failures 70
Procedures for managing disks and volumes 71
Disk management tasks 71
Volume management tasks 75
Chapter 4: Network Administration 79
Using SNMP 80
Configuring the SNMP agent 80
Using the Network Appliance Custom MIB 81
Name resolution 84
The /etc/hosts file 85
Using DNS 87
Disabling DNS 89
Changing your DNS domain name 89
Using NIS 89
Disabling NIS 90
Changing your NIS domain name 91
Displaying the NIS server name 91

Routing 92
Routing table on the filer 92
Specifying default routers 92
Displaying routing status 94
The /etc/dgateways file 94
How the filer replies to requests 95
Using ifconfig to configure an interface 97
ATM PVCs 100
Establishing an outgoing PVC on the filer 100
Establishing an incoming PVC on the filer 101
Establishing a PVC on the remote ATM host 102
Establishing a PVC on interconnecting ATM switches 102
Displaying information about a PVC 102
Deleting a PVC 104

Chapter 5:
Sharing Files between NFS and CIFS Users 105

File-locking interactions 107
Managing symbolic links for CIFS access 108
Enabling symbolic links 108
Redirecting absolute symbolic links 110
Preventing symbolic link cycling 110
NFS and CIFS use of the read-only bit 112
Naming files used by both NFS and CIFS 114
Maximum length of file names 114
Legal characters used in file names 115
Case-sensitivity in file names 115

Timing of directory conversion for CIFS access to NFS-created files 117

Chapter 6: NFS Administration 119

Exporting volumes or directories 120

Exporting to individual clients 120

Exporting to subnets 124

Effects of volume name change on /etc/exports 125

Configuring a filer for (PC)NFS 127

Starting the (PC)NFS server 127

Default umask 129

Configuring a filer for WebNFS 130

Restricting access 130

Turning on WebNFS 130

Specifying and setting the root directory 131

Changing the root directory 131

Disabling the root directory 131

Displaying NFS statistics 132

Chapter 7: CIFS Administration 137

Adding clients to the filer 138

Using cifs commands with rsh 139

Enabling guest access 140

Guest account 140

Generic account 141

Creating a home directory for a CIFS user 142

Managing shares 143
Creating a share 143
Changing the description of a share 144
Displaying information about shares 145
Deleting a share 146
Creating a “home” share for each user 146
Effects of volume name change on CIFS shares 149
Limits on CIFS open files, sessions, and shares 150
Choosing the method for deciding access rights 151
Managing access rights for users or groups 158
Assigning access rights 158
Removing an entry from a share’s ACL 158
Displaying access rights to a file 159
Using CIFS oplocks 160
Using NetBIOS scope IDs 162
Displaying CIFS statistics 163
Displaying CIFS session information 164
Stopping CIFS service 165
Restarting CIFS service 166
Reconfiguring the filer for CIFS 167
Chapter 8: HTTP Administration 169
Starting HTTP service 170
Protecting web pages with passwords 172
The /etc/httpd.access file 172
The /etc/httpd.passwd file 173

The /etc/httpd.group file 173
Web page protection examples 174
Using the HTTP virtual firewall 175
Using virtual hosting 176
Using the /etc/httpd.hostprefixes file 176
Mapping virtual host addresses 177
Specifying MIME Content-Type values 178
Translating URLs 180
Translation rules supported by the filer 180
How the filer processes rules 181
Displaying HTTP connection information 184
Displaying HTTP statistics 185
Chapter 9: Snapshots 187
Snapshot commands 189
Automatic snapshot creation 190
How snapshots work 194
Understanding snapshot disk consumption 196
Disk consumption by multiple identical snapshots 196
How the df command displays snapshots 196
How the snapshot reserve works 197
Managing snapshot disk consumption 199
Scheduling snapshots 200
Displaying snapshot statistics 201
Changing the snapshot reserve 204
Freeing disk space by deleting a snapshot 204

Accessing snapshots from clients 207
Obtaining access to snapshots from an NFS client 207
Obtaining access to snapshots from a CIFS client 208
Determining snapshot versions 210
Determining access times 210
Effects of snapshots on quotas 211
Snapshot options 211
Chapter 10: Managing Quotas and Maximum Number of Files 213
Restricting disk usage by using disk quotas 213
Format of the quotas file 214
The quota command 218
Creating a user quota 222
Creating a group quota 223
Creating a quota tree 224
Removing quota restrictions 225
When quotas are exceeded 226
Increasing the maximum number of files 228
The df command 229
Using the df command with quota trees 230
Chapter 11: Dump and Restore 231
Backing up the filer to tape 233
Features of the filer's dump command 233
Planning for system backups 234
Dump and restore for large volumes 235
Determining the number of tapes required 235
Supported tape drives 238

The dump command syntax 238
Dumping multiple subtrees 243
Restoring dump files 245
Restoring quota trees 246
The restore command 246
The restore command syntax 247
Locally restoring a subtree 249
Restoring the entire filer 251
Examples of restoring from multiple tapes 253
Using a filer tape drive from another system 257
Backing up the filer using NDMP 259
Controlling tape devices 262
The mt command syntax 262
Moving a tape to the end of data 263
Appending a dump 263
Rewinding a tape 263
Taking a tape drive off-line 263
Displaying status information 264
Chapter 12: System Information and Performance 265
Displaying the Data ONTAP version 265
Displaying filer configuration information 266
Displaying disk information using sysconfig -d 266
Displaying RAID information using sysconfig -r 266
Displaying tape drive information using sysconfig -t 266
Displaying overall filer information using sysconfig -v 267
Displaying overall filer information using sysconfig *267*

Displaying volume information 268
Displaying volume state information with vol status 268
Displaying disk information using vol status -d 268
Displaying RAID information using vol status -r 268
Displaying RAID information for each group using vol status -v 268
Displaying filer statistics 269
The sysstat command 269
The uptime command 270
Improving filer performance 271
Displaying network statistics using netstat 273
Chapter 13: Troubleshooting 275
Getting technical assistance 277
Booting from diskette 278
Displaying the boot menu 278
Choosing a boot type 279
Restarting a shutdown filer 280
NVRAM problems 281
NVRAM battery failure 281
Inconsistent NVRAM contents 281
Volume problems 283
Disk problems 286
Disk failure without a hot spare disk 286
Disk failure with a hot spare disk 286
Inconsistent file system 287

Disk operations in maintenance mode 288
Displaying detailed disk information 288
Checking access to a disk 288
Erasing a disk label 288
Configuration problems 289
Filer not accessible from the administration host 289
Damaged or missing /etc/rc file 289
Lost password 291
Initializing all disks and replacing all file systems 291
Network problems 292
NFS problems 294
UNIX cpio problems 296
UNIX df problems 297
Panic messages 298

Glossary 299
Index 307

List of figures

Figure 3-1. Sample disk configuration 61

Figure 3-2. Effects of adding a disk to a file system 62

Figure 4-1. The routed status display 94

Figure 4-2. Output from the arp -a command 103

Figure 9-1. Example of the snap sched command 191

Figure 9-2. How snapshots use disk space 194

Figure 9-3. Directory tree on an NFS client with *vol0* mounted 207

Figure 11-1. Using NDMP to back up data to tape 259

Figure 13-1. Contacting Network Appliance Technical Support 277

Figure 13-2. The boot menu 279

List of tables

Table 2-1. Filer commands you can execute with rsh 9
Table 2-2. Configuration files in /etc 14
Table 2-3. Events that generate automatic email 44
Table 3-1. Usable disk space for various classes of disks 63
Table 4-1. Groups in the custom MIB 82
Table 4-2. Default order for name resolution 84
Table 4-3. Media types for Ethernet interfaces 98
Table 7-1. CIFS limits by filer memory configuration 150
Table 7-2. Absolute mode values 155
Table 7-3. Symbolic mode components 156
Table 9-1. Commands for snapshots 189
Table 11-1. Options for the dump command 240
Table 11-2. Restore command function keys 247
Table 11-3. Restore options 248
Table 13-1. Boot types 279

Preface

About this guide

This guide describes how to configure, operate, and manage Network Appliance® filers that run Data ONTAP™ Release 5.0. It covers all filer models.

The following sections describe how to use this guide.

If you are installing a filer for the first time

Start with the *Quick Start Guide* for concise instructions about setting up a new filer with the default configuration. The information in the *Quick Start Guide* is sufficient for you to prepare the filer for delivering file-sharing service.

If you are upgrading a filer

For instructions about upgrading a filer that is already installed, refer to one of the following documents, depending on which is appropriate to your situation:

- *Upgrade for UNIX*
- *Upgrade for Windows*

Audience

This guide is for system administrators who are familiar with operating systems that run on the filer's clients, such as UNIX, Windows 95, and Windows NT. It also assumes that you are familiar with how the NFS, CIFS, and HTTP protocols are used for file sharing or transfers. It doesn't cover basic system or network administration topics such as IP addressing, routing, and network topology; it emphasizes the characteristics of the Network Appliance filer.

Terminology and conventions

When the term "filer" is used, the information applies to all Network Appliance filer models.

You can enter filer commands on either the system console or from any client computer that can access the filer through `telnet`.

In examples that illustrate commands executed on a UNIX workstation, this guide uses the command syntax of SunOS 4.1.x. The command syntax and output might differ, depending on your version of UNIX.

This guide uses the term "type" to mean pressing one or more keys on the keyboard. It uses the term "enter" to mean pressing one or more keys and then pressing the Enter key.

Keyboard conventions

When describing key combinations, this guide uses the hyphen (-) to separate individual keys. For example, "Ctrl-D" means pressing the "Control" and "D" keys simultaneously. Also, this guide uses the term "Enter" to refer to the key that generates a carriage return, although the key is named "Return" on some keyboards.

The following table describes typographic conventions used in this guide

Convention	Type of information
Italic type	Words or characters that require special attention. File names. Placeholders for information you must supply. For example, if the guide says to enter the `arp -d` *`hostname`* command, you enter the characters "arp -d" followed by the actual name of the host. Man page names. Book titles in cross-references.
`Monospaced` font	Command and daemon names. Information displayed on the system console or other computer monitors. The contents of files.
`Bold monospaced` font	Words or characters you type. What you type is always shown in lowercase letters, unless you must type it in uppercase letters.

Special messages

This guide contains special messages that are denoted by icons in the margin. They are described as follows:



NOTE: A note contains important information that helps you install or operate the system efficiently.



CAUTION: A caution contains instructions that you must follow to avoid damage to the equipment, a system crash, or loss of data.



WARNING: A warning contains instructions that you must follow to avoid personal injury.

Chapter 1: Introducing Network Appliance Filers

The Network Appliance filer is a dedicated, special-purpose server that provides file service to network clients in the UNIX and Windows environments.



NOTE: If you want to quickly set up a new filer with a basic default configuration, refer now to the *Quick Start Guide* provided with the filer documentation set for instructions about how to do this.

Filer features

The following sections list the main features of the filer.

Multiprotocol support

The filer supports the following file-sharing and a file-transferring protocols:

- Network File Service (NFS) for clients running UNIX. The filer also supports the (PC) NFS daemon for clients that run the (PC) NFS protocol.
- Common Internet File System (CIFS) for clients running Windows 95, Windows NT, and Windows for Workgroups.

- Hypertext Transfer Protocol (HTTP), which allows network clients using World Wide Web (WWW) browsers to display files that are stored on the filer.

Administration protocols

The filer supports the following administration protocols:

- Simple Network Management Protocol (SNMP) for network management
- Domain Name Service (DNS) for name resolution
- Network Information System (NIS) for password, group, and netgroup access
- Windows Internet Name Service (WINS) for CIFS name resolution
- Routing Information Protocol (RIP) for default gateway selection
- Remote Magnetic Tape protocol (RMT) to remotely access a filer's tape drive
- The `telnet` protocol for remote logins
- Remote shell (`rsh`) to remotely execute commands on a filer and to use RMT to remotely access tapes on other machines

Disk and file system management features

Filer features that simplify disk and file system management include

- single-volume (file system) default configuration for minimal administrative overhead
- Redundant Array of Independent Disks (RAID) for disk management and protection against data loss due to disk failure

- optional multiple volume and multiple RAID group configuration for filers with large storage capacities
- optional “hot spare” disks, which automatically replace and rebuild the contents of a file system disk in the event one fails
- “hot swap” support, which enables you to add, remove, or replace disks while the system is running
- file system “quotas,” which enable you to restrict disk space usage based on user or group identity
- file system “snapshots,” which simplify backups and enable users to restore their own files without administrator intervention
- Network Data Management Protocol (NDMP) for using third-party backup management programs.
- capability to back up file systems to (and restore from) multiple files while the filer is on-line

What to do next

Use the information in the following table to determine what to do next.

If you want to do this...	See...
Set up a new filer with a default configuration	*Quick Start Guide*
Upgrade an existing filer	*Upgrade for UNIX* or *Upgrade for Windows*
Configure a filer with multiple volumes and RAID groups	Chapter 3, “Managing disks and file systems”

After you set up and configure the filer, refer as needed to the remaining chapters in this book for detailed information about the use and management of the filer features introduced in this chapter.

Chapter 2: Routine Filer Administration

This chapter describes routine filer administration procedures that are needed regardless of the file-sharing protocols licensed for your filer. This chapter emphasizes the filer characteristics that distinguish the filer from a general-purpose server.

For information about advanced filer features, refer to "Chapter 3, "Managing Disks and File Systems," Chapter 9, "Snapshots,"and Chapter 10, "Managing Quotas and Maximum Number of Files."

Using the administration host

During the filer setup procedure, one client computer is specified as the administration host, as described in the *Quick Start Guide*. The following paragraphs describe how you use the administration host:

- The filer grants root permissions to the administration host after the setup procedure in completed. If the administration host is an NFS client, you can mount the filer's root directory and edit a configuration file from the administration host. (You can edit configuration files from any CIFS client as long as you connect to the filer as root. For more information about connecting from a CIFS client to the filer, refer to the *Quick Start Guide*.)

- The filer configures the administration host as the mail host for sending periodic automatic email messages to Network Appliance Technical Support about the status of your filer. For the administration host to send email, it must provide a server for the SMTP protocol, such as the `sendmail` program or Microsoft Exchange server. You can, however, designate another host at your site to be the mail host at any time. Refer to "Automatic email to Network Appliance" on page 43 for more information about how the filer sends automatic email messages. Refer to "Using the options command to configure autosupport" on page 45 for information about how to specify a mail host.

- The administration host is a trusted host from which you can enter filer commands by using a remote shell program such as `rsh`. The filer's setup procedure automatically includes the administration host in the */etc/hosts.equiv* file.

If you plan to manage the filer from an NFS client, use as the administration host a computer that meets the following requirements:

- It supports a text editor that can display and edit text files containing lines ending with the newline character.

- It supports the `telnet` and `rsh` commands.
- It can NFS-mount directories from the filer.

If you plan to manage the filer from a CIFS client, use a computer that supports the `telnet` and `rsh` commands as the administration host.

The root volume

The root volume is the volume that the filer reads configuration files from. Each filer has a root volume, which is specified in one of the following ways:

- If you have only one volume on the filer, that volume is the root volume.
- If you add volumes, you choose a root volume during the multivolume configuration process, described in Chapter 3, "Managing Disks and File Systems."

Volume names begin with the following prefix:

/vol/

Volume concepts are explained in Chapter 3, "Managing Disks and File Systems."

Editing configuration files

Because the filer does not include a local editor, use an editor from a client to change configuration files in the */etc* directory in the filer's root volume. The procedure for modifying configuration files is different depending on whether you edit the files from an NFS client or a CIFS client. If you use an NFS client, edit the files from the administration host. If you use a CIFS client, connect to the filer as Administrator. For more information about how to gain access to the filer from a CIFS client, refer to the *Quick Start Guide*.

From an NFS client

To edit a configuration file from an NFS client, mount the filer's root directory on the administration host and edit the configuration files in the directory from the administration host.



NOTE: You can edit the */etc/exports* file on the root volume from the administration host to give root permission to other clients. Changes take effect only after you invoke the `exportfs` command. After */etc/exports* on the root volume takes effect, you can mount the filer's root directory on these clients and edit the configuration files.

From a CIFS client

Follow these steps to edit configuration files from a CIFS client:

1. Connect from a CIFS client to the filer as Administrator. After the setup procedure, the default */etc/passwd* and */etc/group* files on the root volume enable you to share files on the filer as Administrator.

2. On the CIFS client, display the contents of the filer's *C$* share and select the file to be edited. The filer's root directory is shared automatically as *C$* after setup, and the root has read, write, and execute rights to the share. The *C$* share is a "hidden" share; you can get to it only by specifying the path manually (for example, as *filer**C$*), rather than accessing it through the Network Neighborhood icon.

Obtaining access to the filer shell

The filer supports a command-line interface that you can access directly from the system console or remotely using Telnet. The console and Telnet both provide access to a single session, with everything entered through Telnet being echoed at the console, and vice versa. Only one Telnet session can be open at a time. If you are using Telnet, press Ctrl-] to log out of the filer, then press Ctrl-D to log out of Telnet.

Although Telnet and the console share the same shell session, Telnet and the console each prompt you for a password. Both console and Telnet connections use the same password.

The filer also supports remote shell (`rsh`) with trusted remote hosts (those listed in */etc/hosts.equiv* on the root volume), enabling you to execute some commands on the filer from a shell script or `crontab` file.



NOTE: You can use `rsh` only to invoke filer commands. You cannot use `rsh` to remotely log in to the filer. To log in to a filer from a host, use Telnet.

Table 2-1 lists the filer commands that you can execute using `rsh`.

Table 2-1. Filer commands you can execute with rsh

`cifs`	`mt`	`reboot`	`timezone`
`df`	`netstat`	`restore`	`uptime`
`disk`	`nfsstat`	`route`	`version`
`dump`	`options`	`snap`	`vol`
`exportfs`	`quota`	`snmp`	`ypwhich`
`halt`	`raid`	`sysconfig`	
`httpstat`	`rdate`	`sysstat`	

NOTE: Because the console and Telnet share the same session, entering a Ctrl-C entered aborts whatever command is being run from the console, and vice versa. To make sure that a command is not terminated by an inadvertent Ctrl-C from another station, start the command from a trusted host through `rsh`.

The system password required for establishing a console or Telnet connection with the filer is created during the initial system setup procedure. You can, however, change this password at any time with the `passwd` command, which prompts for the old password, if any, and then requests the new password twice.

For information about other ways to increase filer security besides password protection, use the options described in "Maintaining filer security" on page 50.

Halting and rebooting the filer

As a filer receives requests, it logs them in nonvolatile random-access memory (NVRAM). This feature improves system performance and prevents loss of data in case of a system failure or a power failure. During an orderly shutdown, the `halt` and `reboot` commands flush the contents of NVRAM to disk.

To reboot the filer, use one of the following methods:

- Enter the `halt` command. Proceed as follows, depending on the prompt you get:
  - If the `mon>` prompt appears, enter the `b` command.
  - If the `ok` prompt appears, enter the `boot` command.
- Enter the `reboot` command. The `reboot` command halts the filer and reboots it in one step.

If the filer has a boot diskette in its diskette drive, it boots from the diskette; otherwise, it boots from its hard disk.

The following reasons explain why you should always execute the `halt` command before turning off the filer:

- The `halt` command flushes all data from NVRAM to disk, eliminating a potential point of failure.
- The `halt` command avoids potential data loss on CIFS clients.

  When you halt or reboot the filer, NFS client applications wait for the filer to resume NFS activity. Following reboot, they can resume without error (unless users have canceled them). The effect on CIFS clients is different. If a CIFS client is disconnected from the filer, the users' applications are terminated and changes made to open files since the last save are lost.

  Therefore, never interrupt CIFS service without giving advance warning to users. Before turning off the filer, use the `halt` or

`cifs terminate` command to send a warning message to CIFS users. In this way, the users can save files and exit applications within the time period that you specified before the actual shutdown. For more information about `cifs terminate`, refer to "Stopping CIFS service" on page 165.



NOTE: Clients using Windows 95 or Windows for Workgroups can display the CIFS shutdown messages only when the clients' WinPopup program is configured to receive messages. The capability to display messages from the filer is built into Windows NT.

Understanding the default configuration

The default configuration of a filer depends on whether the filer is running NFS, CIFS, or both, and NIS. Although the default configuration is usable for small sites, it is probably not secure enough for large sites or for sites connected to the Internet. Subsequent sections in this chapter describe in greater detail how you might want to modify the default configuration to suit your needs.

For NFS users, the filer has two exported directories: the root directory and the */home* directory.

For CIFS users, the filer has two shared directories: the root directory and the */home* directory. On CIFS clients, these directories are shown as the *C$* share and the *HOME* share.

The following sections describe the contents of the root and */home* directories.

The root directory

For NFS, the root directory is exported with root permission only to the administration host. From the administration host, you can modify the configuration files in */etc* on the root volume.

For CIFS, you can share the root directory with read and write access on a CIFS client after connecting to the filer. For the connection, use "Administrator" as the user name and the root password specified during the setup procedure.

Table 2-2 lists the configuration files in */etc* on the root volume. Some of the configuration files might not exist on your filer if you use the filer for CIFS or NFS only.

If you are using NIS, the filer uses the NIS group, password, and shadow maps instead of the */etc/group*, */etc/passwd*, and */etc/shadow* files on the root volume.

Table 2-2. Configuration files in /etc

File name	Contents	File-sharing protocol
.cifs.cat	Domain information (only if the filer is a member if a domain)	CIFS only
cifsconfig.txt	CIFS commands that the filer used for configuration	CIFS only
exports	NFS export points	NFS only
hosts	Known hosts and their IP addresses	All
hosts.equiv	Trusted hosts and users for `rsh`	All
group	CIFS group names, GIDs (group identification numbers), and members' names	CIFS
netgroup	Network groups	NFS only
nsswitch.conf	The order in which the filer contacts name services.	All
passwd	Users' names, UIDs (user identification numbers), and primary GIDs	CIFS and (PC)NFS
rc	Script of commands to be executed when the filer is initializing	All

Table 2-2. Configuration files in /etc (continued)

File name	Contents	File-sharing protocol
serialnum	Filer serial number and license codes, which help Network Appliance Technical Support process automatic email sent by the `autosupport` daemon	All
shadow	Encrypted password strings and password aging information	CIFS and (PC)NFS



CAUTION: The filer automatically maintains the */etc/cifsconfig.txt* file on the root volume. Any changes you make to this file are lost after the next `cifs` command. If you edit this file, do so only after you terminate CIFS service by using the `cifs terminate` command. Reboot the filer after you finish editing.

The /home directory

How the */home* directory is used depends on the file-sharing protocol used by the client.

For NFS clients

For NFS clients, the */home* directory is exported with read and write permissions to all NFS clients. Only the administration host has root access. A client can mount a directory on the filer as soon as the filer's host name is added to the client's */etc/hosts* file on the root volume (or to the client's name server).

The svtx bit (also known as the sticky bit) is set on the */home* directory so that users cannot delete each other's files in that directory. This is a safe configuration for networks that use UNIX permissions to provide security for network-accessible files. You can use the `chmod` command from the administration host to clear the sticky bit.

For CIFS clients

For CIFS clients, the */home* directory can be shared by CIFS users as the home share immediately after the setup procedure. However, before users can read and write files in the home share, you must follow the procedures in "Managing access rights for users or groups" on page 158.

By default, CIFS users can write and delete their own files in */home*. For more information about access rights, refer to "Choosing the method for deciding access rights" on page 151

The /etc/rc file

The commands in */etc/rc* on the root volume execute as the system boots to configure the filer. You can execute all the commands in */etc/rc* at the command-line interface. However, if you want to make a change permanent or to make the commands take effect at boot time, you must update the */etc/rc* file on the root volume.

Also, if your filer is licensed to run CIFS, for CIFS to be enabled, */etc/rc* must be present at filer reboot.

The /etc/rc default contents

The best way to understand the commands used in */etc/rc* on the root volume is to examine the following sample */etc/rc* file:

```
#Auto-generated /etc/rc Fri May 30 14:51:36 PST 1997
hostname toaster
ifconfig e0 ‘hostname‘-0
ifconfig e1 ‘hostname‘-1
ifconfig fi0 ‘hostname‘-fi0
ifconfig a5 ‘hostname‘-a5
route add default MyRouterBox
routed on
options dns.domainname netapp.com
options dns.enable on
options nis.domainname netapp.com
options nis.enable on
timezone US/Pacific
savecore
exportfs -a
nfs on
```



NOTE: No CIFS-specific information is entered in */etc/rc* as a result of setup. Do not add CIFS commands in */etc/rc*.

The following list explains the sample */etc/rc* file line by line:

`hostname toaster`
Sets the filer's host name.

```
ifconfig e0 'hostname'-0
ifconfig e1 'hostname'-1
ifconfig f0 'hostname'-f0
ifconfig a5 'hostname'-a5
```
Set the IP address for the filer's Ethernet, FDDI (Fiber Distributed Data Interface), and ATM (Asynchronous Transport Mode) interfaces with a default network mask. (The network mask is set during the setup procedure, as described in the *Quick Start Guide*.) The arguments in single backquotes expand to "toaster" if you specify "toaster" as the host name during setup. The actual IP addresses are obtained from the */etc/hosts* file on the root volume; you might prefer to enter IP addresses directly in */etc/rc* on the root volume. If the specified network interface is not present, `ifconfig` issues an error message and has no other effect.

The actual interface names and numbers depend on the filer model. Refer to "Naming conventions for network interfaces" on page 22 for more information about interface names.

If you change the filer's host name, you must modify the */etc/hosts* file on the root volume to substitute the new host name. If you don't, `ifconfig` fails.

To override the default network mask, explicitly specify the network mask in the `ifconfig` command after the host name; for example:

```
ifconfig e1 'hostname'-1 netmask 255.255.0.0
```

`route add default LocalRouter 1`
Specifies the default router. You can add route commands to */etc/rc* on the root volume to set static routes for the filer. The network address for LocalRouter must be in */etc/hosts* on the root volume.

`routed on`
Starts the routing daemon. See “Routing” on page 92 for more information about routing.

```
options dns.domainname netapp.com
options dns.enable on
```
These options set the DNS domain name and enable DNS.

```
options nis.domainname netapp.com
options nis.enable on
```
These options set the NIS domain name and enable NIS.

`timezone US/Pacific`
Sets the time zone. The argument to the `timezone` command specifies which file in the */etc/zoneinfo* directory on the root volume describes the time zone you want. A list of time zone names is in the *Quick Start Guide.*

`snap sched 0 2 6@8,12,16,20`
Sets the schedule for automatic snapshot creation. The line in the example */etc/rc* file is the default setting, which creates and keeps no weekly snapshots, two daily snapshots (created at midnight every night), and six hourly snapshots (created at 8 a.m., noon, 4 p.m., and 8 p.m.). Chapter 9, “Snapshots,” describes snapshots in more detail.

`savecore`
Saves the core file from a system panic, if any, in the */etc/crash* directory on the root volume. Core files are created only during the first boot after a system crash.

`exportfs -a`
Exports all directories specified in the */etc/exports* file on the root volume. This command is included only if the filer runs NFS.

`nfs on`
Turns on NFS file service. This command is included even if NFS is not licensed for your filer. When the filer runs this command from */etc/rc* on the root volume on a filer without an NFS license, the command fails and the following messages appear:

```
NFS service is not licensed.
(Use the “license” command to license it.)
NFS server is NOT running.
```

`dns on`
Turns on DNS.

`nis on`
Turns on NIS.

Editing /etc/rc

To update or make changes to */etc/rc*, follow the instructions in “Editing configuration files” on page 7. Network Appliance recommends that you make a backup copy of the file before you start editing. After you finish editing, reboot the filer. The reboot tests the new configuration immediately.

Changing SNMP commands in /etc/rc

Regardless of whether SNMP is enabled at your site, add the following commands to */etc/rc* on the root volume:

- **snmp contact *“your email and telephone number”***
- **snmp location *“location of your filer”***
- **snmp init 1**

For example:

```
snmp contact “jdoe@abc.com 415-555-1212”
snmp location “ABC corporation, engineering lab”
snmp init 1
```

The information provided by these commands enables Network Appliance Technical Support to contact you after receiving automatic e-mail messages about your filer from your site. For information about automatic email messages, refer to "Automatic email to Network Appliance" on page 43.

Naming conventions for network interfaces

For information about how network interfaces are numbered, refer to the hardware guide for your filer model. This section discusses how software refers to each network interface.

When you write or modify shell scripts that involve interface names, remember that different interface naming conventions are used for different filers:

Filers use the following interface naming conventions:

- Ethernet interface names start with the letter “e” followed by the slot number (for example, e0) and the port number, if the adapter card is a quad-port Ethernet card (for example, e1a and e1b).
- FDDI interface names start with the letter “f” followed by the slot number (for example, f1 and f2).
- ATM interface names start with the letter “a” followed by the slot number (for example, a5).

Filers automatically create host names for network interfaces by appending the slot number of the interface to the filer’s host name. For example, if the filer is named toaster, the host name for e0 is toaster-0, and the host name for e1a is toaster-1a. If the interface is an FDDI interface, the letter “f” precedes the slot number. For example, the host name for f1 is toaster-f1.

Recovering from configuration errors

Because the filer doesn’t have a local text editor, recovering from certain configuration errors might require more steps than modifying the configuration files.

For example, because of an incorrect IP address in */etc/rc* on the root volume, the filer might have become inaccessible to clients. To solve this problem, follow these steps:

1. Enter commands on the console to configure the interface with the correct address
2. Export the configuration file to the administration host
3. Edit the file from the host
4. Reboot the filer.

If configuration errors render the hard disk unbootable, try booting from the floppy disk. For more information, see "Booting from diskette" on page 278.

Core files

The filer might crash if there is a hardware or software failure. On the first boot after a system crash, the filer stores the core file in */etc/crash* on the root volume. The savecore command, which is included in */etc/rc* on the root volume by default, produces two core files, *core.n-small* and *core.n* (the *n* in the file name is a number). The *core.n-small* file contains everything except the file system buffers. The *core.n* file includes more information that might be essential for determining the cause of the crash; it is often necessary for investigating problems involving the file system.

The savecore command also displays a message on the system console and logs a message in */etc/messages* on the root volume.

If you followed the instructions for system setup in the *Quick Start Guide*, your filer sends email automatically to Network Appliance Technical Support upon each system reboot. However, if you disabled automatic email, contact Network Appliance Technical Support when your system creates a core file.

Message logging

The message logging daemon `syslogd` and the configuration file, */etc/syslog.conf* on the root volume specify how system messages are logged. Based on their severity and origin, you can direct system messages to the console, to a file, or to a remote system. By default, all system messages (except for those of severity level debug) are written to the console and to the */etc/messages* file on the root volume.

The */etc/syslog.conf* file consists of lines with two tab-separated (not space-separated) fields of the following form:

`facility.level action`

The following sections describe each parameter:

The facility parameter

The facility parameter specifies the subsystem from which the message originated. It is one of the following keywords:

`auth`	messages from the authentication system, such as login
`cron`	messages from the internal `cron` facility
`daemon`	messages from filer daemons, such as `rshd`
`kern`	messages from the filer kernel
`*`	specifies all facilities

The level parameter

The level parameter describes the severity of the message. The level is one of the following keywords (higher to lower severity):

`emerg`	panic condition that causes a disruption of normal service

`alert`	condition that you should correct immediately, such as a failed disk
`crit`	critical conditions, such as disk errors
`err`	errors, such as those caused by a bad configuration file
`warning`	conditions that might become errors if not corrected
`notice`	conditions that are not errors, but might require special handling
`info`	information, such as the hourly uptime message
`debug`	used for diagnostic purposes (suppressed by default)
*	specifies all levels

The action parameter

The action parameter specifies the place to forward the received message when it originates from the specified facility and is of the specified level or higher. This field can be one of the of the following:

- the pathname of a file beginning with a slash (for example, */etc/messages*)
- a remote host beginning with the @ sign (for example, *@adminhost*)
- the console (*/dev/console* or an asterisk)

For example, if you specify *err.kern* and */etc/messages* on a line, all kernel messages of levels `emerg`, `alert`, `crit`, and `err` are forwarded to the */etc/messages* file.

Every Sunday at midnight, the */etc/messages* file is copied to */etc/messages.0*, the */etc/messages.0* file is copied to */etc/messages.1*, and so on. The system saves messages for up to six weeks.

Check the */etc/messages* file once a day for important messages. You can automate checking this file by creating a script on the administration host that periodically searches */etc/messages* and then alerts you.

The following example shows a customized */etc/syslog.conf* file:

```
# Log anything of level info or higher to /etc/messages.
*.info                          /etc/messages

# Log all kernel messages, and anything of level err or
# higher to the console.
*.err;kern.*                    /dev/console

# Log all kernel messages and anything of level err or
# higher to a remote loghost system called adminhost.
*.err;kern.*                    @adminhost

# Log messages from the authentication system of level notice
# or higher to the /etc/secure.message file. This file has
# restricted access.
auth.notice                     /etc/secure.message
```

For more information about `syslog.conf`, see the *na_syslog.conf*(5) man page.

Configuring filer options

You configure filer options with the `options` and `vol options` commands.

The options command

When used interactively, the `options` command displays option values or changes the filer's behavior temporarily; the system returns to the default state when rebooted. To make changes permanent, you must include the `options` commands in the */etc/rc* file.

The syntax of the `options` command is as follows:

```
options [ option [value] ] ...
```

where:

option is the name of the option.

value is the value of the option.

For example, the command to enable the `pcnfsd` daemon is as follows:

```
options pcnfsd.enable on
```

If you do not supply a value, the command displays the current value of the specified option.

The vol options command

You use the `vol options` command to configure volume-level behavior. You can use this command only with volume options, which are listed in "Volume options" on page 39. The `vol options` command has the following syntax:

`vol options` *`volname option`* [*`value`*]

where:

`volname` is the name of the volume that the option applies to.

`option` is the name of the option.

`value` is the value of the option.

For example, to set the maximum size of a RAID group in volume *myvol* to 12, enter:

```
vol options myvol raidsize 12
```

If you do not supply a value, the command displays the current value of the specified option in the specified volume.

Detailed option information

Options are described in detail in other sections of this guide and in the man pages. The *na_options*(1) man page contains information about `options` command options. The *na_vol*(1) man page contains information about the `vol options` command options.

The sections that follow provide a brief description of each option grouped by function. The default value is below the option name.

Option values

The following conventions apply to default values listed:

- "None" means that there is no default value.
- If the default is "on," the other possible value is "off."
- If the default is "off," the other possible value is "on."

The values for on and off are not case-sensitive. If you do not supply a parameter in the `options` command, the command prints the current values of all available options.

Autosupport options

These options control whether and how the filer sends automatic status messages. For more information about autosupport, see "Automatic email to Network Appliance" on page 43.

autosupport.doit
None (default)

Immediately sends an email message describing the status of the filer. A word entered as the value for the option is sent in the notification subject line and should describe the reason for the notification.

autosupport.enable
On (default)

Enables the autosupport daemon, which sends automatic email messages to report the status of the filer.

autosupport.from
autosupport (default)

Specifies the sender of the automatic email message. It should be the email address of the person whom Network Appliance contacts after receiving an automatic email message.

autosupport.mailhost
administration_host (default)

Specifies the mail hosts that receive automatic email messages. Use a comma-separated list with no spaces.

autosupport.noteto
None (default)

Specifies up to five recipients of an automatic short email message. Use a comma-separated list with no spaces.

autosupport.to
autosupport@netapp.com (default)

Specifies up to five recipients of autosupport email messages. Use a comma-separated list with no spaces.

CIFS options

These options control CIFS features on the filer.

cifs.generic_account
None (default)

When you set`cifs.generic_account` to the name of an account, users who have an account in a trusted domain but not an entry in */etc/passwd* or NIS can use that account. All generic users have the same rights to the generic users' account.

cifs.guest_account
None (default)

When `cifs.guest_account` is blank, a CIFS user can log in to the filer without an account in the password database, provided a domain controller authenticates the user.

When you set `cifs.guest_account` to the name of an account in the NIS `passwd` map or */etc/passwd* (typically guest), guest access to the filer is enabled, and the user has the UNIX user ID and the group ID of the guest account. For more information, see "Enabling guest access" on page 140.

cifs.home_dir
None (default)

Specifies the complete path name of the "homes directory." The directories under this path should have the names of users as their names. When a CIFS user connects to the filer and there is a directory name that exactly matches the user's name, the user sees a share of that name (truncated to 12 characters) that is the user's home directory. Only the user can access the home directory using this share. All other users are denied access. For more information, refer to "Creating a home directory for a CIFS user" on page 142.

cifs.oplocks.enable
On (default)

When this option is on, the filer enables clients to use oplocks (opportunistic locks) on files. Oplocks provide a significant performance enhancement, but have the potential to cause lost cached data on some networks with impaired reliability or latency, particularly wide-area networks. In general, you should disable this option only if there are problems with databases and to isolate problems.

cifs.scopeid
(blank) (default)

Specifies a second element for a single-element NetBIOS computer name. This element is case-sensitive. You use this option to isolate a group of computers on a network that communicate only with other computers with the identical NetBIOS Scope ID.

Microsoft recommends that you not use NetBIOS Scope IDs.

Do not use this option if you are using DNS on your network.

For more information, refer to "Using NetBIOS scope IDs" on page 162.

cifs.symlinks.cycleguard
On (default)

If an object being accessed by a CIFS client is a symbolic link, the `cifs.symlinks.cycleguard` option, when set to ON, eliminates the possibility of cyclic directories. It does so by preventing the following of symbolic links that contain the "dot" (".") or "dot-dot" ("..") component—symbolic links that could refer to a directory higher in the same tree. With the `cifs.symlinks.cycleguard` option set to Off, if you are careful, you can use symbolic links having "dot" or "dot-dot" components. For more information about this option, refer to "Preventing symbolic link cycling" on page 110.

cifs.symlinks.enable
On (default)

When you set `cifs.symlinks.enable` to On (the default setting), if the object being accessed by a CIFS client is a symbolic link, the filer follows the link with the condition that the ultimate target turns out to reside within the originating share. This ensures that the client has access permission to the target. This applies both to relative symbolic links (links to paths beginning with a character other than "/" and treated as a path relative to the parent directory of the symbolic link) and absolute symbolic links (links to paths beginning with "/" and treated as a path relative to the root of the file system). For more information about this option, refer to "Managing symbolic links for CIFS access" on page 108.

DNS options

These options control how the filer works with DNS. For more information about DNS, see Chapter 4, "Network Administration."

dns.domainname
None (default)

Sets the DNS domain name to the specified domain name.

dns.enable
Off (default)

Enables the DNS client on the filer. Before you enable DNS, you must set the DNS domain and the */etc/resolv.conf* file must exist.

HTTP options

These options enable and control HTTP services. For more information about HTTP on the filer, see Chapter 8, "HTTP Administration."

httpd.admin.enable
On (default)

Enables HTTP access to the filer's on-line help files and other files used by FilerView.

httpd.enable
Off (default)

Enables the HTTP server.

httpd.log.max_file_size
2147483647 (2 GB - 1 byte) (default)

Specifies the number of bytes */etc/log/httpd.log*, the HTTP log file, can grow to. The maximum value is 4294967295 (4 GB - 1 byte).

httpd.rootdir
None (default)

Specifies the root directory containing files and directories that HTTP transfers to clients.

httpd.timeout
900 (15 minutes) (default)

Specifies the minimum amount of time, in seconds, before an idle HTTP connection times out.

httpd.timewait.enable
On (default)

When you set this option to On, the filer drops an HTTP connection one minute after the client closes it. When you set this option to Off, the connection is not dropped and resources are consumed until the connection times out.

NFS and (PC)NFS options

These options to enable and control NFS. For more information about NFS, see Chapter 6, "NFS Administration."

nfs.mount_rootonly
On (default)

Configures the filer to give mount privilege to root using only privileged ports (ports 1 through 1,024).

nfs.per_client_stats.enable
Off (default)

Specifies whether the filer collects and displays NFS statistics from individual clients.

nfs.tcp.enable
Off (default)

Specifies whether the filer supports NFS over TCP. Enable this option if a client has problems using NFS over UDP.

nfs.v2.df.2gb.lim
Off (default)

Limits to 2 GB the response the filer gives to requests regarding total space, free space, or available space. This option is necessary for some NFS clients to calculate the amount of free space accurately. Without this option, a file system with more than 2 GB of free disk space might appear to be full to the client that initiates the "file system statistics" request.

nfs.v3.enable
On (default)

Specifies whether the filer supports NFS Version 3. Disable this option if a client has problems using NFS Version 3 and that client cannot be configured to use NFS Version 2.

nfs.webnfs.enable
off (default)

Turns WebNFS on and off.

nfs.webnfs.rootdir
None (default)

The specified directory becomes the root or public directory for webNFS. When a request specifies a relative path, lookups for files are done with respect to this directory.

nfs.webnfs.rootdirset
FALSE (default)

When set to TRUE, sets the directory specified in the `nfs.webnfs.rootdir` option to be the webNFS root or public directory.

pcnfsd.enable
Off (default)

Specifies whether the filer runs the (PC)NFS server.

pcnfsd.umask
022 (default)

The default is 022, which means that users can read and write their own files, but members of their group and others can only read the files.

NIS options

These options control how the filer works with NIS. For more information about NIS, see Chapter 4, "Network Administration."

nis.domainname
None (default)

Sets the NIS domain to the specified domain name.

nis.enable
Off (default)

Enables the NIS client on the filer. You must set the NIS domain before you enable NIS.

RAID options

These options control how the filer uses RAID. For more information about RAID on the filer, see Chapter 3, "Managing Disks and File Systems."

raid.reconstruct_speed
4 (default)

Specifies the speed of RAID reconstruction. The speed ranges from 1 (slowest) to 10 (fastest). The filer uses the number to determine the percentage of CPU time used for RAID reconstruction.

raid.scrub.enable
On (default)

Specifies whether the filer performs RAID scrubbing.

raid.timeout
24 (default)

Sets the time, in hours (from 1 through 24), that the system runs in degraded mode before an automatic shutdown.

Volume options

These options control volume-level operations. They are different from other options because you use them only with the `vol options` command, described in "The vol options command" on page 29.

minra
Off (default)

Configures the filer to perform minimal read ahead. By default, the option is disabled and the filer does aggressive read ahead.

no_atime_update
Off (default)

Prevents the update of the access time (atime) on an inode when a file is read. This option prevents inode updates from contending with reads from other files. Use it only on a filer with extremely high read traffic (for example, on a news server used by an Internet access provider or on a filer used mainly as an HTTP server).

nosnap
Off (default)

Temporarily disables automatic snapshots.

nosnapdir
Off (default)

Makes invisible the snapshot directory that's usually present at the client mount point or at the root of the CIFS share. It also turns off access to the snapshot directory and all snapshot directories under the mount point or the root of the CIFS share.

After you toggle this option, you might not notice the effect immediately because the information about the snapshot directories might still be in the client's attribute cache. To force the change to take effect immediately, unmount and remount the file system.

raidsize
none (default)

Sets the maximum size of a RAID group in *volume*. Must be an integer greater than zero.

root
none(default)

Makes *volume* the root volume.

Miscellaneous options

These options control additional aspects of filer operation.

ip.path_mtu_discovery.enable
On (default)

Enables or disables path MTU discovery, which is currently used only by TCP. When enabled, it enables the filer to discover and use the largest packet size that the filer can send to another host without fragmenting a packet. This means that the filer doesn't have to limit itself to sending many small packets, which takes more time and resources than sending fewer large packets.

If you cannot establish a connection, set this option to Off.

rps.status
None (default)

When you set this option to Good, clears the message that a power supply in the RPS (redundant power supply) is bad. Set this option to Good after you replace a failed power supply in the RPS; otherwise, you keep getting the message even though the power supply is no longer bad. With no value, this option reports the status of the filer's RPS.

telnet.hosts
All hosts (default)

Specifies a list of hosts that can log in to the filer using Telnet. You can limit Telnet access to up to five specified hosts. The hosts should be listed in a comma-separated list. You can disable Telnet for all hosts by specifying a hyphen (-).

wafl.maxdirsize
10240 (default)

Sets the maximum size, in kilobytes, of a directory file. A directory file with a size of 10240 kilobytes can hold about 300,000 files.

wafl.root_only_chown
On (default)

Enables only the root user to change the owner of a file. When you disable the option, the owner of a file can change its ownership without being root. By default, this option is enabled.

When a nonroot user changes the owner of a file, the set-user-id and set-group-id bits are cleared. If a nonroot user tries to change the owner of a file but the change would cause the file's recipient to exceed his or her quota, the attempt fails.

Automatic email to Network Appliance

This section describes the `autosupport` daemon, which controls how automatic email messages are sent from your site to Network Appliance Technical Support.

How the autosupport daemon works

The `autosupport` daemon is enabled by default on the filer. The daemon triggers automatic email messages to members of Network Appliance Technical Support, alerting them to potential filer problems. If necessary, Network Appliance contacts you at the email address that you specify to help resolve a potential system problem. Refer to "Using the options command to configure autosupport" on page 45 for more information about specifying the email address and other options.

Because the filer doesn't function as a mail host, it relies on another host at your site that listens on the SMTP port (25) to send mail. Therefore, `autosupport` requires at least one host reachable by the filer that runs an SMTP server or a mail forwarder, such as the `sendmail` program or Microsoft Exchange server. By default, the administration host defined during system setup is used as a mail host. You can specify more mail hosts.

The mail host sends email to Network Appliance Technical Support about your filer after any of the events listed in Table 2-3.

You can specify up to five addresses of email recipients. The default recipient is Network Appliance Technical Support. For receiving automatic messages, the email address of Network Appliance Technical Support is autosupport@netapp.com. Refer to "Using the options command to configure autosupport" on page 45 for more information about autosupport.

Table 2-3. Events that generate automatic email

Event	Subject line of the email message
Low NVRAM lithium battery	BATTERY_LOW!!!
Disk failure	DISK_FAIL!!!
Disk scrub occurred	DISK_SCRUB!!!
Fan failure	FAN_FAIL!!!
Shutdown because of overheating	OVER_TEMPERATURE_SHUTDOWN!!!
Partial RPS failure[a]	POWER_SUPPLY_DEGRADED!!!
System reboot	REBOOT
Disk shelf error	SHELF_FAULT!!!
Weekly backup of */etc/messages*	WEEKLY_LOG
`option autosupport.doit` command	THE STRING SPECIFIED IN OPTION `autosupport.doit`

a. This email message is sent only by a filer that uses an RPS. It means that one of the two power supplies has ceased functioning.

Contents of automatic email messages

Each email message generated by `autosupport` contains the following types of information:

- date and time stamp of the message
- NetApp software version
- system ID of the filer
- host name of the filer
- software licenses enabled for the filer

- SNMP contact name and location (if specified in */etc/rc*)
- output of the following commands (some are applicable only to the licensed protocols):

```
sysconfig -v
options
ifconfig -a
nfsstat -c
cifs stat
cifs sessions
cifs shares
httpstat
df
df -i
snap sched
sysconfig -r
```

- contents of */etc/messages*
- contents of */etc/serialnum*

Using the options command to configure autosupport

To change the default behavior of the autosupport daemon, use the options command. As with other filer commands, you can add options commands to the */etc/rc* file if you want to execute them automatically when the filer reboots.

Disabling or enabling the autosupport daemon

The autosupport daemon is enabled by default. The syntax of the command to disable or enable the daemon is as follows:

```
options autosupport.enable on|off
```

For example:

```
options autosupport.enable off
```

Specifying mail hosts

The command to specify hosts that send autosupport email messages to Network Appliance Technical Support is as follows:

```
options autosupport.mailhost hostname,...
```

You can specify up to five mail host names. Separate names by commas and do not include spaces in the list. The default is the administration host. For example:

```
options autosupport.mailhost host1,host2,host3
```

Specifying addresses for autosupport mail

The command for specifying the recipients of automatic email messages sent by the autosupport daemon is as follows:

```
options autosupport.to addresses,...
```

You can specify up to five email addresses. Separate email addresses by commas and do not include spaces in the list. The default is autosupport@netapp.com. For example:

```
options autosupport.to autosupport@netapp.com,d@c.com
```

Be sure to enter the command on a single line.

Specifying the filer administrator's address

Sometimes Network Appliance Technical Support might need to contact the administrator of the filer after receiving automatic email. The options command for specifying the filer administrator is as follows:

```
options autosupport.from address
```

For example:

```
options autosupport.from jdoe@abc.com
```

If Network Appliance Technical Support needs to determine whether further action is necessary for solving or prevent filer problems, it sends email to jdoe@abc.com.

Sending an immediate message

The command to send an automatic email message immediately is as follows:

```
options autosupport.doit string
```

The string is used in the subject line of the email message to explain why the email was sent. For example:

```
options autosupport.doit TESTING
```

Immediate messages contain the same filer data as automatic messages.

Sending a short message

The `options autosupport.noteto` command specifies the recipients of short email messages sent by autosupport. The short email messages are for urgent events, such as disk failures or filer reboots. The following example shows a short message:

```
Return-Path: <autosupport>
Received: from toaster by netapp.com (4.1/SMI-4.1)
id AA14370; Thu, 26 Sep 96 07:51:31 PDT
Message-Id: <9609261451.AA14370@netapp.com>
From: autosupport
To: jdoe
Date: Thu, 26 Sep 1996 07:51:27 -0700
Resent-Date: Thu, 26 Sep 1996 7:58:42 PDT
Subject: System Alert from toaster
REBOOT on Thu Sep 26 07:51:27 PDT 1996
```

Short messages are useful if the person who should be notified of urgent events reads email on a small screen, such as the screen on an alphanumeric pager.

Synchronizing filer system time

The filer uses two commands to control the system time. The `date` command sets the time locally, while the `rdate` command synchronizes the filer's time with the time on another host.

Setting the date and time

Set the date and time. For example:

```
toaster> date 199705220925
```

sets the date and time to 9:25 a.m. on May 22, 1997.

Synchronizing the filer

The `rdate` command synchronizes the filer's time with a target computer's time using the UDP time service (typically, UDP port 37). This service is available on most UNIX computers. Check */etc/services* and */etc/inetd.conf* on the target computer to see whether this service is supported.



NOTE: There is currently no Windows 95 or Windows NT counterpart to the `rdate` command. If you want to use the `rdate` command, you must have a UNIX workstation that supports `rdate` on your filer's network.

To keep the filer's clock accurate, regularly run the `rdate` command on the filer with the target machine, a computer that maintains accurate time and that supports the port 37 UDP time service. For example, if the name of the target computer is "time_node," enter

```
toaster> rdate time_node
```

A typical scheme is to have a UNIX computer run a periodic `cron` job that executes the appropriate `rdate` command on your filer through `rsh`. The computer and the user (if not root) running the `cron` job must be in the filer's */etc/hosts.equiv* file on the root volume.

For example, on the UNIX named called adminhost, the user named adminuser sets up a cron job to run every day at 3 a.m. It directs the filer named toaster to request a time update from the computer named time_node, which maintains an accurate time and supports the UDP time service.

The crontab entry on the UNIX system is as follows:

```
sys% 0 3 * * * rsh toaster -l root rdate time_node
```

The */etc/hosts.equiv* file on toaster must contain the following line:

```
adminhost adminuser
```

The *adminhost* and *time_node* host names must be known to the filer.

Maintaining filer security

The following options to the `options` command help maintain filer security:

- The `telnet.hosts` option restricts Telnet access to a limited number of hosts. Use this option to specify a comma-separated list of up to five hosts that can log in to the filer using Telnet. Alternatively, you can disable Telnet for all hosts by specifying a hyphen (-). By default, the option argument consists of an asterisk (*), which means all hosts have Telnet access.

  A record of all Telnet and console logins is maintained in the */etc/messages* file on the root volume.

- The `mount_rootonly` option restricts the mount privilege to root using privileged ports (ports 1 through 1,024). By default, the option is enabled. This option accepts the arguments `off` and `on` to disable and enable it. This option is applicable only if your filer runs NFS.



NOTE: Some PC clients and some older implementations of NFS on UNIX workstations use nonprivileged ports to send requests. If you have these clients at your site, disable the `mount_rootonly` option or upgrade the client software.

- The `wafl.root_only_chown` option enables only root to change the owner of a file. When the option is disabled, the owner of a file can change its ownership without being root. By default, this option is enabled. Use the arguments `off` and `on` to disable and enable the option.

  When a nonroot user changes the owner of a file, the set-user-id and set-group-id bits are cleared. If a nonroot user tries to change the owner of a file but the change causes the file's recipient to exceed his or her quota, the attempt fails. This option is applicable only if your filer runs NFS.

Software licenses

The filer requires software licenses to enable NFS, CIFS, or HTTP services. Licenses are installed on the filer at the factory per your order, so the initial setup of your filer does not involve entering license codes.

You need to enter license codes only if any of the following conditions applies:

- You purchased a filer with a release earlier than Release 4.0, and you are upgrading it to Release 4.0 or later.
- You want to enable a protocol not previously licensed for your filer.
- The filer's file system becomes corrupt and must be rebuilt.

Network Appliance provides you with the appropriate license codes when shipping you the software upgrade kit or when giving you instructions for obtaining the software upgrade over the Internet.

Enabling NFS, CIFS, or HTTP service

To enter a license code to enable a protocol on your filer, use the license command with the following syntax:

`license` *`protocol=code`*

For example, if license code for NFS is "ABCDEFG," enter

`license nfs=ABCDEFG`

The events that take place after a license command depend on the protocol specified:

NFS — The filer displays the following messages:

```
nfs license enabled.
        nfs enabled.
```

The filer also automatically runs the `nfs on` command to start NFS service. However, the filer does not add the `nfs on` command to */etc/rc* as a result of the license command. If you want the filer to run NFS service after each reboot, add `nfs on` to */etc/rc*.

CIFS — The filer displays the following messages:

```
CIFS license enabled.
Run cifs setup to enable CIFS.
```

To start CIFS service, set up the filer's CIFS configuration by running `cifs setup`. You don't need to run `cifs setup` after the license command if you already set up the CIFS configuration as described in the *Quick Start Guide*.

HTTP — The filer displays the following messages:

```
http license enabled.
Use "options httpd.enable" to enable http.
```

To start HTTP service, enter the following command:

```
toaster> options httpd.enable on
```

The `options` command takes effect immediately. If you want the filer to automatically start HTTP service after each reboot, add `options httpd.enable` to */etc/rc*.

The following example shows how to activate several protocols with one command:

```
license nfs=ABCDEFG CIFS=HIJKLMN http=PQRSTUV
```

Displaying current license codes

To display licensing information, enter the `license` command without parameters as follows:

```
toaster>license
nfs=ABCDEFG
cifs=currently not licensed
http=PQRSTUV
```

Disabling a license

To disable a license, enter the DISABLE as the code for the protocol. For example, to disable your filer's license for NFS, enter

```
license nfs=DISABLE
```

After your disable a license, your filer stops service for the corresponding protocol. You can restart the service by reentering the license code.

Replacing license codes

If you misplace a license code, contact Network Appliance Technical Support to obtain a copy.

Chapter 3: Managing Disks and File Systems

This chapter provides information about how to manage the filer's disks and file systems. It is divided into two main sections:

- Filer fundamentals
- Procedures for managing disks and volumes

Filer fundamentals

This section explains filer disk and file system concepts and principles. It describes how the filer manages disk resources and organizes the data stored on the filer. It is organized into two main sections:

- Disk concepts
- Volume concepts

Disk concepts

This section covers the following topics:

- Understanding RAID groups
- Understanding hot spare disks
- Understanding hot swap
- Using disks of various sizes
- Usable space on each disk
- Handling disk failures

Understanding RAID groups

The filer uses RAID Level 4 to ensure data integrity even when one of the disks fails. The file system design, together with the support for RAID, optimizes filer performance and enables you to incrementally expand the filer's disk storage capacity.

In a RAID group, different disks have different functions. Most of the disks in the RAID group are *data disks*. One disk is the *parity disk*, which enables the filer to recover the data on a data disk if one fails.

Beginning with Release 5.0, the filer supports multiple RAID groups. The factory default filer configuration contains one RAID group. The filer supports up to 32 RAID groups. Each RAID group belongs to only one volume; you cannot assign more than one volume to the same RAID group.

Spare disks are used by RAID groups as needed. They do not have to be in the same disk shelf to be available to a RAID group. The filer automatically assigns disks to RAID groups and creates new RAID groups as each RAID group is filled with its maximum number of disks.

About RAID group size

The following characteristics apply to RAID group size:

- The default number of disks in a RAID group (including the parity disk) is 14.
- A RAID group must contain at least two disks.
- The largest RAID group size you can create *manually* is 28 disks.
- The maximum RAID group size is 52 disks.

Occasionally, you might encounter situations in which you want to specify a RAID group size other than the default. For example, you might want to configure a filer with smaller RAID groups for the following reasons:

- Using smaller RAID groups reduces disk reconstruction time if a disk fails.
- Using smaller (therefore, more) RAID groups provides higher reliability by reducing the risk of data loss due to double- or multiple-disk failure.

Conversely, configuring with larger (therefore, fewer) RAID groups in a filer uses fewer disks for parity, leaving more disks available for data storage

About parity disks

In each RAID group, the filer assigns the role of parity disk to the largest disk in the RAID group. After a data disk failure, the filer uses the parity disk in conjunction with the other data disks to reconstruct the failed disk's data and optionally write it onto a *hot spare disk*. The parity disk must be at least as large as the largest data disk.

For more information about how the filer recovers from disk failures by using the parity disk or hot spare disk, refer to "Handling disk failures" on page 63.

Disk address

You use the disk address to

- interpret screen messages (for example, command output or error messages) that you see on your display
- quickly locate the disk that the message is referring to

You identify a disk by its address, which is listed in the `HA.Disk_ID` column of the output of the `sysconfig -r` command. In this output listing, "`HA`" refers to the host adapter and "`Disk_ID`" refers to the disk ID number.

In the case of a SCSI disk, the disk address is a combination of the disk's SCSI adapter number and the disk's SCSI ID number on the adapter. For example, disks attached to SCSI adapter 0 are numbered 0.0 through 0.6, disks attached to SCSI adapter 9a are numbered 9a.0 through 9a.6, and so on.

To locate the ID for a fibre channel disk drive referenced in the display output, look for the drive label on the disk shelf.

For more information about locating specific drives on disk shelves, refer to the hardware guide for the disk shelf.

RAID scrubbing

The filer uses the RAID scrubbing procedure to increase data availability. The filer scans each disk in the RAID group for media errors. If the filer finds media errors, it fixes them by reconstructing the data from parity and rewriting the data.

Disk scrubbing reduces the chance of a multiple disk failure, caused by a disk media error encountered while the system was running in degraded mode.

The filer only scans a RAID group when all of the group's disks are operational. Although disk scrubbing slows the filer somewhat, network clients might not notice the change in the filer's performance because disk scrubbing starts automatically at 1:00 A.M. on Sunday, when most systems are lightly loaded.

By default, disk scrubbing is enabled. You might want to disable scrubbing if you have a recurring problem that scrubbing encounters. For example, there might be an unrecoverable error on a disk that you cannot fix before the next disk scrub. The following commands disable and enable disk scrubbing:

```
options raid.scrub.enable off
options raid.scrub.enable on
```

You can also manually start and stop disk scrubbing regardless of the current value (on or off) of the `raid.scrub.enable` option. Following are the commands for starting and stopping scrubbing manually:

```
disk scrub start
disk scrub stop
```

Messages from the disk scrubbing process are sent to the system error logging daemon. Following are sample messages that can appear.

If the filer finds an inconsistent parity block during scrubbing, it prints the following messages:

```
Inconsistent parity on volume volume_name, RAID group
n, stripe #n.
Rewriting bad parity block on volume volume_name, RAID
group n, stripe #n.
```

If the filer finds a media error on the parity disk, it prints the following message:

```
Rewriting bad parity block on volume volume_name, RAID
group n, stripe #n
```

If the filer finds a media error on a data disk, it prints the following message:

```
Rewriting bad block from parity on disk n, block n
```

If the filer finds more than one bad block, it prints the following message:

```
Multiple bad blocks found on volume volume_name, RAID
group n, stripe #n
```

The following sample messages appear after disk scrubbing is complete:

```
Scrub found n parity inconsistencies
Scrub found n media errors
Disk scrubbing finished...
```

Understanding hot spare disks

In addition to data disks and parity disks, the filer supports zero or more hot spare disks. A hot spare disk is not part of any RAID group and does not contain file system data. After a disk failure, the filer automatically rebuilds data (or parity) onto a hot spare disk, which then replaces the failed disk in the RAID group. This procedure avoids a system shutdown and returns the system to full performance.

A hot spare disk cannot replace a failed disk that is larger than itself. If you use only one hot spare disk in a filer, the hot spare disk should be as large as the largest file system disk. If you have multiple hot spare disks installed, the system uses the smallest hot spare disk needed to replace the failed disk.

Understanding hot swap

The filer enables you to *hot swap* disk drives. Hot swapping a disk drive means installing or removing it from the disk shelf while the filer is running, with minimal interruption to a file system. You might want to hot swap a disk into a filer to replace a disk or to add a hot spare disk.

Using disks of various sizes

You can use disks of various sizes in the same filer. Figure 3-1 shows a sample configuration.

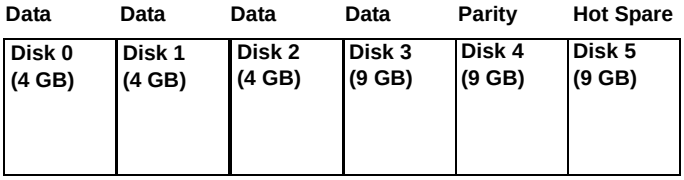


Figure 3-1. Sample disk configuration

The following sections discuss the guidelines for choosing the size of a disk to add to the filer or to use as a replacement.

When replacing a failed disk

When you replace a failed disk, the physical space of the new disk must be equal to or larger than the *usable* space of the replaced one. The Used column of the `sysconfig -r` output shows the amount of usable space on each disk.

However, if the physical space of the new disk is larger than the usable space of the old disk, the system can use no more space than was available on the old disk. For example, if you replace a 4 GB disk with a 9-GB disk, the extra 5-GB on the new disk is not available to the file system. Therefore, it is most cost-effective to replace a disk with a disk of the same size.

When adding a disk to a file system

When you add a disk to a RAID group that is larger than the current parity disk, the new disk automatically becomes the parity disk. The disk used as the parity disk before the disk addition becomes a data disk. Figure 3-2 illustrates how the disk configuration changes after you add a hot spare to a file system.

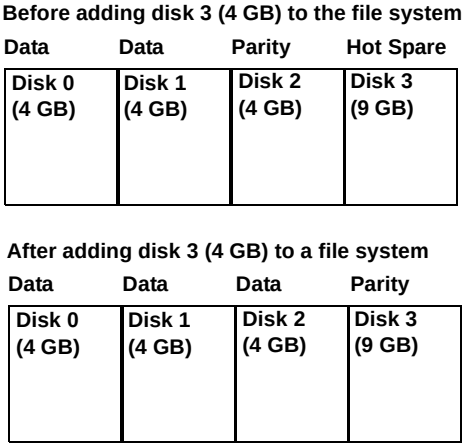


Figure 3-2. Effects of adding a disk to a file system

Because the hot spare disk (Disk 3) is larger than the parity disk when you add it to the system, the filer automatically uses it as the parity disk. Disk 2, which was previously the parity disk, becomes a data disk.

Usable space on each disk

A disk's usable space can be different from its physical space. The information here applies to data, parity, and hot spare disks.

Disks from different manufacturers might differ slightly in size even though they belong to the same size category.

To ensure that you can replace one disk with another in the same size category, a filer shipped with release 2.1 or later automatically "right-sizes" its disks. That is, it rounds off the disk capacity and divides disks into 2 GB, 4 GB, and 9 GB classes.

If you upgraded your filer from a release before 2.1, the filer doesn't right-size the existing disks. Instead, it continues to use all the physical disk space.

"If a disk has never been right-sized, you can only replaced it with another disk that *has the same number of blocks or more.*

Table 3-1 shows the usable disk space for each class of disk supported by the filer.

Table 3-1. Usable disk space for various classes of disks

1-GB disks	2-GB disks	4-GB disks	9-GB disks
1000 MB (2048000 blocks)	2000 MB (4096000 blocks)	4000 MB (8192000 blocks)	9000 MB (18432000 blocks)

Although automatic disk right-sizing is not applied to existing disks in an upgraded system, it is applied to disks being added to the filer.

You can use `sysconfig -r`, to compare the physical space and the usable space. You can also determine whether disks are right-sized.

Handling disk failures

If one block on a data disk fails, the filer uses the parity disk for its RAID group to reconstruct the data on that block. The block is mapped to a new location on disk. If an entire data disk fails, the parity disk for that RAID group prevents any data loss and allows the filer to continue running

Although the filer can continue to function with a failed disk, if it cannot reconstruct that failed disk onto a hot spare, it automatically shuts down after 24 hours to encourage you to replace the failed disk. You can change the amount of time from 24 hours to another value using the `raid.timeout` option to the `options` command.

The effects of a disk failure on filer operation depend on whether the filer has a hot spare disk.

Without a hot spare disk

If the filer is not equipped with a *hot spare disk*, after a disk fails, the filer enters a state called "degraded mode. In this state, the RAID feature allows the filer to continue to run without losing data (although the filer's performance is affected). Replace the failed disk as soon as possible because a second disk failure in the same RAID group causes the entire file system to be lost.

When a disk fails, the filer logs a warning message in the */etc/messages* file and to the system console every hour, notifying you of the number of hours before the system shuts down.

The shutdown ensures that you notice the disk failure. You can restart the filer without fixing the disk, but the filer continues to shut itself off at the specified intervals until you repair the problem.

By default, the filer shuts down after 24 hours. You can change this time interval using the `option raid.timeout` command; the argument is the time in hours that the system runs before automatic shutdown.

The system shuts down after the specified period if it is running in degraded mode. A filer is in degraded mode if either of the following conditions exist:

- one disk in any RAID group has failed
- the batteries on the NVRAM card are low (if the filer is a PCI-based system)

With a hot spare disk

If you reserve one or more disks as hot spare disks when you configure your filer, the filer also enters degraded mode after a disk failure. However, the filer immediately begins rebuilding the missing data in the background on the hot spare disk, with minimal interruption to file service. The filer logs this activity in the */etc/messages* file and does not automatically shut down. If you

turn off the filer while it is in degraded mode, it stops data reconstruction. After you turn the filer back on, the filer restarts the data reconstruction process from the beginning.

Except for a loss in performance while data is rebuilt on the hot spare disk, the failure of a single disk is transparent to the user. The filer exits degraded mode and returns to normal operation after it finishes reconstructing the data.

Network Appliance recommends that you replace the failed disk with a new hot spare disk after the filer finishes reconstructing data. This way, the filer continues to have a hot spare disk that it can use in case another disk fails.

The `sysconfig -r` command displays which disk is reserved as the hot spare disk. In addition to disk failure and hot spare disk replacement activity, the */etc/messages* file logs any failure in a periodic check of the hot spare disk.

RAID data reconstruction speed

You can control the speed of RAID data reconstruction by entering the following command:

```
options raid.reconstruct_speed speed
```

where *speed* is a number ranging from 1 (slowest) to 10 (fastest). Because RAID data reconstruction consumes CPU time, sometimes increasing the speed of reconstruction slows the filer's network operations. The default speed is 4, which means approximately 40% of the CPU time is used for RAID data reconstruction.

When RAID data reconstruction is in progress, use the `sysstat` command to check the system load on the filer. If the load is light, increase the speed of RAID data reconstruction to maximize CPU utilization. For more information about the `sysstat` command, refer to "Displaying filer statistics" on page 269.

Volume concepts

This section covers the following topics:

- Understanding volumes
- Determining the number of volumes to use
- Planning a multiple volume configuration
- About foreign volumes
- Handling volume failures

Understanding volumes

Data on the filer is organized in *volumes*. A volume is an independent file system with its own RAID groups.

The initial configuration for new filers running Data ONTAP Release 5.0 includes one 2 to 14 disk volume (a root file system). All remaining disks are spares.

About volume naming conventions

You choose the volume names. The names must follow these naming conventions:

- begin with either a letter or an underscore (_)
- contain only letters, digits, and underscores
- contain no more than 255 characters

About the root volume

Each filer must have a *root volume* to boot. The root volume of a filer (configured with either a single or multiple volumes) is the volume whose */etc* directory is used by the filer for configuration information.

The filer uses two naming conventions to indicate the root volume: */vol/vol0* and */*.

In the */vol/vol0* convention, *vol0* is the default name of the root volume for a filer initially installed with Release 5.0. You can change this name using the `vol rename` command. The */vol* prefix indicates that the next part of the path (such as *vol0* in this example*)* is a volume name.

For backward compatibility with filers running releases of Data ONTAP earlier than 5.0, you can use the */* naming convention to indicate the root volume on a filer that you upgraded to Release 5.0 that contains a previously existing file system.

Mounting volumes

On filers configured with multiple volumes, mounting */* is equivalent to mounting */vol/vol0/*, (where *vol0* is the root directory of the root volume). Paths that begin with */* (for example, */etc*) refer to directories on the root volume.



NOTE: */vol* is not a directory— it is a special virtual root path under which the filer mounts other directories. You cannot mount */vol* to view all the volumes on the filer; you must mount each filer volume separately.

In mount requests and server commands, prefix the path names of the volumes and directories you want to mount using the convention */vol/volume_name/directory*. For example, */vol/users/home/cheryl* is a directory called */home/cheryl* in a volume named *users*.

Determining the number of volumes to use

Whether you should use the default single volume configuration or create additional volumes depends mainly on the storage capacity of the filer.

If you want the filer to be configured with a single volume, you do not need to do any further volume configuration after you complete the initial setup. You can add more disks to an existing volume after you have created it.

If your filer doesn't have a large number of disks, a single-volume configuration is probably all you need. You can create additional volumes in the future.

Using multiple volumes or RAID groups requires additional configuration beyond the default setup. If you decide to configure the filer with additional volumes, the following sections provide instructions for planning configurations that use multiple volumes and RAID groups on your filer.

Pros and cons of configuring with multiple volumes

Configuring with multiple volumes aids in the administration of filers that have large storage capacities, enabling you to

- perform administrative and maintenance tasks (for example, backup and restore) on individual volumes rather than on a single, large file system
- set `option` command values (for example, `snap sched`, `raidsize`, `minra`, `no_atime_update`, and so on) differently for individual volumes
- take individual volumes off-line (for example, to perform administrative tasks on their file systems or associated RAID groups) while the others remain on-line, without interrupting the availability of the data on them

Limitations of configuring with multiple volumes include

- The filer's storage space is partitioned.
- Additional administrative overhead is introduced (for example, defining export points).
- You can expand but not concatenate, shrink, or split volumes.

- You cannot perform a local copy of a volume's contents (you must use `dump` and `restore or ndmp copy`).

Planning a multiple volume configuration

If you want to set up a filer with multiple volumes, read this section before you configure the volumes.

For instructions about specific volume configuration procedures, see "Volume management tasks" on page 75.

Deciding the number and sizes of volumes you want

Before you configure a filer with multiple volumes, decide on the number and sizes of volumes you want to configure.

When deciding the number and sizes of volumes you want to configure, keep the following considerations in mind:

- Configuring with more volumes
  - provides more flexible quota and snapshot configuration
  - requires more export points
- Configuring with larger volumes increases the time needed to restore a volume from tape.
- The maximum number of volumes per filer is 23.
- The maximum *recommended* volume size is 250 GB.

About foreign volumes

You can remove an entire volume from one filer and install it in another. For example, you might want to move a volume to a different filer to

- replace the volume's disk shelf with one that has a greater storage capacity
- gain access to the files on a dead server

When a filer detects a foreign volume at boot time, it places the foreign volume off-line. The foreign volume can then be brought on-line.

Handling volume failures

A volume might fail due to an *inconsistent* directory or a double-disk failure. If the system does not reboot after a volume failure, take the volume off-line, as follows:

1. Boot the filer from a boot diskette into maintenance mode.
2. Use the `vol` command to take the failed volume off-line.

If the failed volume was the filer's root volume, you must designate another volume as the new root volume.

Procedures for managing disks and volumes

This section provides step-by-step procedures for performing many filer management tasks from the command line on a filer administration host (or client). It is organized into two main sections:

- Disk management tasks
- Volume management tasks

Alternatively, you can perform these procedures using the FilerView graphical interface.

Disk management tasks

This section contains procedures for managing the filer's disks and RAID groups.

Setting the size of a volume's RAID groups

To set the RAID group size of a volume when you create it, enter the **`vol create`** ***`volume`*** **`-r`** ***`n`*** command, where ***`volume`*** is the name of the volume and ***`n`*** is the number of disks you want in each RAID group. Every RAID group must contain at least two disks.

Changing the size of a RAID group after creating it

To change the size of a RAID group, enter the **`vol`** ***`volume`*** **`options raidsize`** ***`size`*** command, where ***`size`*** is the number of disks you want in the RAID group.



NOTE: You can only change the size of the last RAID group in a volume. You cannot change the size of RAID groups after they have been filled.

Installing a new disk

A new disk is one that has never been used. To install a new disk, follow these steps:

1. From the console, enter the **`disk swap`** command.

2. Install the disk according to the hardware guide for your disk shelf.

   The system displays a message confirming that a disk was swapped, then waits 15 seconds as the disk is turned on. The system recognizes the disk as a hot spare disk. (You can verify that the new disk has been added by checking the output of the command **sysconfig -r**.)

Adding disks to volumes

1. Enter the **sysconfig -r** command to verify that there are spare disks available for you to add.

2. Add the disks to a volume by entering **vol add *volume* *ndisks***, where ***volume*** is the name of the volume and ***ndisks*** is the number of disks you want to add to the volume.

Refer to the *na_vol(1)* man page for details about adding disks to volumes.

Installing multiple disks

To install multiple disks, you can follow the procedure in "Installing a new disk" page 71 for each disk, or you can follow these steps:

1. Shut down the filer with the `halt` command.

2. Turn off the filer.

3. Install the disk drives into the disk shelf according to the instructions for your disk shelf.

4. Turn on the filer.

If another filer used the disks, or if the filer displays an error message, erase the disk label before the filer boots with the disks installed.

CAUTION: If the disk contains a disk label that was previously used in another filer's file system, the filer halts during the normal boot process. If this occurs, refer to the next section, "Erasing a disk label," for instructions about how to proceed.

Erasing a disk label

To erase a disk label, boot the filer from a system boot diskette into maintenance mode, then use the `disk_erase_label` command as described in "Disk problems" on page 286.

Removing a failed disk

If a disk fails, enter the **`disk swap`** command before removing the failed disk. Then follow the instructions in the hardware guide for your filer model to remove the disk.

Removing a hot spare disk or an active file system disk

You might want to swap disks because you want to use the disk in another filer (if the disk is a hot spare) or because a disk is logging excessive errors.

If the disk is operational, follow these steps to swap the disk:

1. If the disk is a hot spare, go to step a. If the disk is a file system disk, go to step b.

   a. Enter the **`disk remove`** command to remove a hot spare disk. Use the `sysconfig -r` command to determine the disk number to enter in the command. The `disk remove` command spins down the disk. After it stops spinning, the disk is ready to be removed. Go to step 2.

   b. Enter the **`disk fail`** command to remove a file system disk. Use the `sysconfig -r` command to determine the disk number to use in the command. The `disk fail` command spins down the disk. After the `disk fail` command, the system operates in degraded mode, which means that a disk is missing from the RAID group. Go to Step 2.



NOTE: The `disk fail` command permanently marks a disk as failed. You cannot reuse the disk; you must be replace it.

2. Enter the `disk swap` command.

   The `disk swap` command stalls all file activity until you complete the Step 3.

3. Remove the disk from the filer following the instructions in the hardware guide for your filer model. File service resumes 15 seconds after you remove the disk.

Handling disk errors during hot swap

If the filer detects a hung disk during a hot swap operation, it displays a warning message and then shuts down. A system reboot usually clears transient disk errors that happen during the hot swap. Follow these steps to recover from those errors:

1. Disconnect the disk shelf used during the hot swap from the power source and wait 15 seconds to allow all drives to spin down.

2. Reconnect the power source to the disk shelf.

3. At the system monitor prompt, enter **b** to reboot the system.

After the reboot, your change to the disk array takes effect.

Contact Network Appliance Technical Support if you encounter any disk problems during a hot swap. Be prepared to provide information about your system, such as the output from the `sysconfig -v` command, messages in */etc/messages*, and precisely how you performed the hot swap.

Adding or removing a hot spare disk on a filer that doesn't support hot swap

On filer models that don't support hot swapping, follow these steps to add or remove a hot spare disk:



CAUTION: The system might hang if you do not install the disk properly.

1. Verify that the hot spare disk is at least the same size as the largest disk currently in any RAID group.
2. Use the `halt` command to shut down the filer.
3. Turn off the filer.
4. Install or remove the hot spare disk according to instructions in the hardware guide for your filer model.
5. Reboot the system. The system automatically recognizes the added disk as a hot spare disk.

Volume management tasks

This section contains procedures for configuring and managing volumes.



NOTE: Although you can expand volume sizes and the RAID groups assigned to them after you create an initial multiple volume configuration, you cannot split or shrink volumes after you create them.

Creating volumes

You can create up to 23 volumes on a filer. Each volume must contain at least two disks.

To create a new volume

- Enter **`vol create`** ***`newvol n`*** at the system prompt, where ***`newvol`*** is the name for the new volume and ***`n`*** is the number of disks to use. You must have at least ***`n`*** spare disks available.

After you create a new volume on a CIFS filer, you must create shares that refer to the new volume to enable clients to access it.

After you create a new volume on an NFS filer, you must

1. Update the system */etc/exports* file.
2. Run `exportfs`.
3. Add the appropriate mount point information to the */etc/fstab* or */etc/vfstab* file on clients that mount volumes from the filer.

Adding disks to a volume

To add more disks to an existing volume

- Enter **`vol add`** ***`vol n`***, where ***`vol`*** is the name of the volume and ***`n`*** is the number of disks to be added.

Monitoring volume status

To determine volume status (such as size, options, disk assignments, and so on)

- Enter the **`vol status`** ***`volume`*** command.

To see the RAID group and individual disk information for a particular volume

- Enter the **`vol status -r`** ***`volume`*** command.

To see the RAID group and individual disk information for all volumes

- Enter the **`vol status -r`** command.

Setting volume options

To set various volume options

- Enter the **`vol options`** ***`volume option value`*** command.

Making a volume inactive

To remove a volume from active use upon next reboot

- Enter the **vol offline *volume*** command.

Reactivating an off-line volume

To reactivate an off-line volume

- Enter the **vol online *volume*** command.

Adding a foreign volume

To add a foreign volume (that is, one moved from another filer)

1. Follow the instructions in the hardware guide to remove the disk shelf from the old filer.
2. Turn off the destination filer and install the disk shelf in it.
3. Turn on and boot the destination filer.

   When the destination filer boots, it places the foreign volume off-line.
4. Enter **vol online *volume*** to bring the volume on-line in the new filer.

Destroying a volume

To destroy a volume, turning its disks back into spare disks

1. Enter the **vol offline *volume*** command to deactivate the volume.
2. Enter the **reboot** command to reboot the filer.
3. Enter the **vol destroy *volume*** command to destroy the volume.

Renaming a volume

To rename a volume

- Enter the **vol rename *oldvolume newvolume*** command.

- Update the */etc/exports* file and run `exportfs`.

Update any CIFS shares that refer to the volume.

Chapter 4: Network Administration

This chapter discusses the following topics:

- how to use SNMP on the filer from a UNIX host, including information about the Network Appliance custom Management Information Base (MIB)
- how the filer resolves names
- how the filer routes packets using its routing table
- how to configure a network interface
- how to set up a virtual firewall
- how to work with ATM (Asynchronous Transfer Mode) PVCs (permanent virtual circuits)

Using SNMP

For diagnostic and other network management services, the filer supports the SNMP MIB-II specification. Based on SNMP version 1, this specification provides data about the following MIB-II groups:

- system
- interfaces
- address translation
- IP
- ICMP
- TCP
- UDP
- SNMP

MIB-II. SNMP commands enable users to specify up to eight communities and trap notifications for up to eight management stations.

Configuring the SNMP agent

To use SNMP, configure the SNMP agent using the `snmp` command. A typical set of SNMP commands in the */etc/rc* file in the root volume is as follows:

```
snmp contact 'jdoe@abc.com 415-555-1212'
snmp location 'ABC corporation, engineering lab'
snmp community add ro private
snmp traphost add snmp-mgr1
snmp init
```

SNMP commands that NetApp supports are described in detail in the *na_snmp*(1) man page. The following paragraphs provide brief explanations of these commands:

`snmp contact 'jdoe@abc.com 415-555-1212'`
Sets the email address and telephone number of the person responsible for the filer. You can include the person's full name, but an email address and a telephone number enable Network Appliance Technical Support to contact the right person after receiving an automatic email message through autosupport from your site concerning your filer status.

`snmp location 'ABC corporation, engineering lab'`
Sets the physical location of the filer. This value is returned by the SNMP agent.

`snmp community add ro private`
Creates a read-only community called private.The SNMP manager uses the community name as a password to communicate with the filer's SNMP agent.

`snmp traphost add snmp-mgr1`
Makes the system snmp-mgr1 the recipient of all SNMP traps from the filer.

`snmp init 1`
Initializes the SNMP daemon with the values specified with `snmp` commands. This command also sends the SNMP cold start and links up or down traps, as appropriate, to any trap hosts that were previously registered using the `snmp traphost` command. This command should be the last SNMP command in the filer's */etc/rc* file.

Using the Network Appliance Custom MIB

The Network Appliance custom MIB provides detailed information about many aspects of filer operation. If your filer was shipped with Release 4.0 or later, the MIB is in the */etc/mib/netapp.mib* file in the root volume. If you upgraded your filer from a pre-4.0 release, the */etc/mib/netapp.mib* file does not exist on your filer. You can obtain the custom MIB from the NOW web page at *http://wwwnow.netapp.com*. Install this MIB file on your Network Management Station according to the installation

procedure for your station, so that your station can obtain information from the filer about the objects that are part of this MIB.

New Network Appliance custom MIB

The Network Appliance custom MIB has been updated to version 1.1.0. It contains new objects that help you manage multivolume features.

Using deprecated single-volume objects

Single-volume objects are deprecated, but you can use them for single-volume systems as before. If you are using a single-volume filer, you do not need to make changes to use the new MIB.

Finding multivolume objects

The descriptions of the deprecated single-volume objects contain the names of the corresponding new multivolume objects. For multivolume objects, use the new objects rather than the deprecated ones.

The top-level groups in the custom MIB and the information they contain are described in Table 4-1.

Table 4-1. Groups in the custom MIB

Group name	Contents
product	Product-level information, such as the software version string and the system ID.
sysstat	Some system-level statistics, such as CPU uptime, idle time, and the number of kilobytes transmitted and received on all network interfaces.
nfs	Statistics like those displayed by the `nfsstat` command, including statistics for each client if per-client statistics are enabled. The per-client statistics are indexed by client IP addresses.

Table 4-1. Groups in the custom MIB (continued)

Group name	Contents
quota	Information related to disk quotas, including the output of the `quota report` command. To access quota information, quotas must be turned on. For more information about quotas, see "Restricting disk usage by using disk quotas" on page 213.
filesys	Information related to the file system, including the equivalent of the `maxfiles` and `df` commands, and some of the information from the `snap list` command. For more information about `maxfiles`, see "Increasing the maximum number of files" on page 228. For more information about the `df` command, see "The df command" on page 229. For more information about snapshots, see Chapter 9, "Snapshots."
raid	Information about RAID equivalents to the `sysconfig -r` output. For more information about `sysconfig`, see Chapter 12, "System Information and Performance."

Name resolution

The filer resolves host names by searching maps (or databases) for services to use. The filer tries name resolution services in a default order or in the order that you specify in the */etc/nsswitch.conf* file in the root volume. For example, by default, first the filer tries to resolve host names locally by searching the */etc/hosts* file in the root volume (designated by "files" in Table 4-2 and in the */etc/nsswitch.conf* file in the root volume). If the name is not resolved, it tries NIS (if NIS is enabled). If the name is still not resolved, the filer requests services from a DNS server (if DNS is enabled). You can specify any or all of the resolution methods.

Table 4-2 shows the default search order for each map.

Table 4-2. Default order for name resolution

Map	Services in search order
`hosts`	files, NIS, DNS
`passwd`	files, NIS
`netgroup`	files, NIS
`group`	files, NIS
`shadow`	files, NIS

To specify a different order in which the filer contacts name services, create the */etc/nsswitch.conf* file in the root volume. Each line must have the following format:

`map``: `*`service ...`*

`map` is one of the following maps or databases: `hosts`, `passwd`, `netgroup`, `group`, and `shadow`.

service is one or more of the following:

- files for local files in the */etc* directory in the root volume
- dns for DNS
- nis for NIS

List services in the order in which you want the filer to contact the services. For example, the following file instructs the filer to contact first NIS for hosts, then DNS, and finally local files in */etc* in the root volume. For passwords, the contact order is NIS, then local files in */etc* in the root volume.

```
hosts: nis dns files
passwd: nis files
```

When a name is resolved, the search stops.



NOTE: When performing CIFS operations, the filer can use WINS servers for name service. However, filer commands unrelated to CIFS use the */etc/hosts* file, DNS, or NIS to resolve names, as described in the following sections.

The /etc/hosts file

The filer can use the */etc/hosts* file in the root volume to resolve names used in the */etc/rc, /etc/syslog.conf, /etc/dgateways, /etc/exports, /etc/netgroup, and /etc/hosts.equiv* root volume files. If you do not use any host names, */etc/hosts* is not needed.

The sample setup session shown in the *Quick Start Guide* generates the following root volume */etc/hosts* file:

```
#Auto-generated by setup Wed Jun 4 19:06:18 PST 1997
192.9.200.100   toaster-0 toaster
192.9.201.100   toaster-1
192.9.202.100   toaster-f0
192.9.200.1     MyRouterBox
192.9.200.2     Adminhost
192.9.200.2     Mailhost
```

By default, the filer reads the */etc/hosts* file in the root volume whenever it needs to resolve host names, so changes to the file take effect immediately. If you change the IP address for an interface, the new IP address doesn't take effect until you reboot the filer and the appropriate `ifconfig` command is executed from */etc/rc*. It is safest to reboot the filer after changing any */etc* file to make sure that the new configuration works correctly.

If you use DNS and you put files first in the */etc/nsswitch.conf* file in the root volume, the filer looks up host addresses in */etc/hosts* in the root volume before sending queries to the DNS server. Therefore, it is important that each entry in */etc/hosts* contains accurate information. To reduce the need for updating information in */etc/hosts* in the root volume, when using DNS, keep only a minimum number of entries in */etc/hosts*.

If you use NIS, you can do one of the following actions:

- Modify the Makefile of the NIS master to copy the NIS master's */etc/hosts* file from the root volume to the filer when it is changed.
- Have the filer use NIS directly.

For example, you can put the following line at the end of the NIS Makefile section for *hosts.time*:

```
@mntdir=/tmp/netapp_etc_mnt_$$$$;\
if [ ! -d $$mntdir ]; then rm -f $$mntdir; mkdir $$mntdir; fi;\
for filer in toaster1 toaster2 toaster3 ; do \
mount $$filer:/etc $$mntdir;\
mv $$mntdir/hosts $$mntdir/hosts.bak;\
cp /etc/hosts $$mntdir/hosts;\
umount $$mntdir;\
done;\
rmdir $$mntdir
```

Substitute the name of each filer in the "for" list in place of toaster1, toaster2, and so on.

Using DNS

The filer includes DNS client capabilities. This means that it can query DNS servers for host-name-to-IP-address and IP-address-to-host-name translation services. With DNS enabled, you no longer have to update the filer's */etc/hosts* file in the root volume every time you add a new host to the network. (If you use the default search order or put files before DNS in the */etc/nsswitch.conf* file, you still have to update the */etc/hosts* file if one of its entries changes before the filer tries to resolve host names.)

To prevent naming inconsistencies, Network Appliance recommends that when you enable DNS, you use only the default */etc/hosts* file in the root volume.

Enabling DNS during system setup

At system setup, if you enter **y** in response to the following prompt, the setup program prompts you for a DNS domain name as follows:

```
Do you want to run DNS resolver [n]: y
Please enter DNS domain name []:
```

After you enter a DNS domain name, setup prompts you for the IP addresses of up to three DNS name servers. Based on the IP addresses you enter, setup generates the */etc/resolv.conf* file in the root volume. Entries in */etc/resolv.conf* file consist of the word "nameserver" followed by an IP address, as follows:

```
nameserver ip_address
```

For details about name server query policies, see the *na_resolv.conf*(5) man page.

Enabling DNS without using setup

If you didn't start DNS during the system setup process, you can start DNS using the following procedure:

1. Create an */etc/resolv.conf* file in the root volume. The file consists of up to three lines, each specifying a name server host in the following format:

   ```
   nameserver ip_address
   ```

   For example:

   ```
   nameserver 192.9.200.10
   nameserver 192.9.200.20
   nameserver 192.9.200.30
   ```

2. Edit the */etc/rc* file in the root volume to make sure that the option specifying the DNS domain name is set and that the option to enable DNS is set to on. For example:

   ```
   options dns.domainname netapp.com
   options dns.enable on
   ```

3. For the commands in step 2 to take effect, reboot the filer or enter the commands at the filer prompt.

DNS is now enabled, and you no longer have to update the filer's */etc/hosts* file in the root volume every time you add a new host to the network (unless you specify files first in the */etc/nsswitch.conf* file in the root volume).

Disabling DNS

To disable DNS, enter the following command or put the command in the */etc/rc* file in the root volume to make the change permanent:

```
options dns.enable off
```

Changing your DNS domain name

To change your DNS domain name, enter the following command or put the command in the */etc/rc* file in the root volume to make the change permanent:

```
options dns.domainname domainname
```

Using NIS

The filer includes NIS client capabilities. This means that it can query NIS servers for host-name-to-IP-address and IP-address-to-host-name translation services.



NOTE: Because the `nsswitch.conf` file already enables you to specify the order in which the filer finds password information, you do not need to use + or - entries in the filer's */etc/passwd* file in the root volume. Any existing + or - entries are ignored.

The filer uses the following NIS maps:

```
hosts.byname
hosts.byaddr
passwd.byname
passwd.byuid
passwd.adjunct
group.byname
group.bygid
netgroup.byhost
```

Shadow password information is obtained from the `passwd.byname` map.

Enabling NIS during system setup

During system setup, the following prompt appears:

```
Do you want to run NIS Client [n]:
```

If you enter **y**, setup prompts you for an NIS domain name, as follows:

```
Please enter NIS domain name [] :
```

Enter an NIS domain name.

Enabling NIS without using setup

If you didn't start NIS during the system setup process, you can start NIS using the following procedure:

1. Edit the */etc/rc* file in the root volume to make sure that the option specifying the NIS domain name is set and the option to enable NIS is on. Insert lines similar to the following:

   ```
   options nis.domainname netapp.com
   options nis.enable on
   ```

2. Enter the `options` commands from the command line or reboot the filer. Options entered only on the command line are not saved if you reboot.

NIS is now enabled, and you no longer have to update the filer's */etc/hosts* file in the root volume every time you add a new host to the network.

Disabling NIS

To disable NIS, enter the following command or put the command in the */etc/rc* file in the root volume to make the change permanent:

```
options nis.enable off
```

Changing your NIS domain name

To change the domain name on a running system, enter the following command or put the command in the */etc/rc* file in the root volume to make the change permanent:

```
options nis.domainname new.domain
```

Displaying the NIS server name

To display the NIS server name, enter the following command:

```
ypwhich
```

The `ypwhich` command has no options.

Routing

Even though the filer can have multiple network interfaces, it does not function as a router; that is, the filer does not route packets between its interfaces on behalf of other network hosts.

Routing table on the filer

For routing its own packets, the filer relies on the default route and explicit routes. Typically, the filer learns explicit routes through `icmp` redirect messages received from the default router; you do not need to enter explicit routes in the filer's routing table. To display the filer's current routing table, use the `netstat -r` command. For example:

```
toaster> netstat -r
Routing tables
Internet:
Destination      Gateway            Flags Refs   Use  Interface
default           nwo-               UG    1      138  e0
298.295.227       link#1             UC    0        0  e0
298.295.227.1     link#1             UHL   1       24  e0
nwo-              8:0:20:79:f9:79    UHL   1        0  e0
298.295.227.255   link#1             UHL   1     3696  e0
```

If it is necessary to enter explicit information into the filer's routing table, use the `route` command. See the *na_route*(1) man page about how to add or modify information in the routing table.

If the filer cannot find an explicit route in the routing table for a particular destination, it uses the default route. This means that the filer sends the traffic to the default router, which is specified in the */etc/dgateways* file in the root volume.

Specifying default routers

One default router is specified during initial system setup. You can, however, add potential default routers at any time to the */etc/dgateways* file in the root volume. For each added router, you should also specify the metric, which is a number indicating the

preference for the router. The highest preference is 1, which is also the default preference for the router specified during system setup. The lowest preference is 15. For information about the format of the */etc/dgateways* file, refer to "The /etc/dgateways file" on page 94.

To help manage multiple routers and to enable you to create redundant routing schemes, the filer runs the `routed` daemon, a simple routing daemon, which is enabled at boot time. This daemon "listens" for Routing Information Protocol (RIP) packets being exchanged between routers on the network to determine which routers are alive. From the routers that are alive, the `routed` daemon selects the one with the highest preference to use as the filer's default router.

However, the filer doesn't rely on the `routed` daemon to construct the routing table. The function of the filer's `routed` daemon is to check the status of the default router. Refer to the *na_route*(1), *na_routed*(1), and *na_dgateways*(1) man pages to learn more about routing on the filer.

You turn the `rrouted` daemon off and on with the `routed` command.

To turn the `routed` daemon off, enter:

```
toaster> routed off
```

To turn the `routed` daemon back on, enter:

```
toaster> routed on
```

If you turn the `routed` daemon off by editing */etc/rc* in the root volume, manually designate a default router in */etc/rc*.

Displaying routing status

To display the status of the default gateway list, use the `routed status` command. The `-n` option forces the command to display numeric values for gateway names. An example of the `routed status` display is shown in Figure 4-1.

```
toaster> routed status
  RIP snooping is on
  Gateway   Metric State   Time Last Heard
  karl           1  ALIVE  Wed Mar 9 03:38:41 GMT 1994
  groucho        1  ALIVE  Wed Mar 9 03:38:41 GMT 1994
  292.0.266.366  1  ALIVE  Wed Mar 9 03:38:41 GMT 1994
  292.0.266.377  1  ALIVE  Wed Mar 9 03:38:41 GMT 1994
  june           2  ALIVE  Wed Mar 9 03:38:41 GMT 1994
  292.0.266.332  2  ALIVE  Wed Mar 9 03:38:41 GMT 1994
  292.0.266.352  3  ALIVE  Wed Mar 9 03:38:41 GMT 1994
  292.0.266.351  4  ALIVE  Wed Mar 9 03:38:41 GMT 1994
  292.0.266.350  5  ALIVE  Wed Mar 9 03:38:41 GMT 1994
```

Figure 4-1. The routed status display

The `routed status` display shows the following information:

- whether RIP snooping is active
- the current list of default gateways
- the metrics of the default gateways
- the state of the gateways (ALIVE or DEAD)
- the last time each gateway was heard from

The /etc/dgateways file

The */etc/dgateways* file in the root volume is the configuration database for the `routed` daemon. From the routers that `routed` has determined to be alive, `routed` selects the one with the highest preference to be the default router. When the filer cannot find an explicit route for a packet, it routes the packet to the default router.

The file consists of lines with the following format:

gateway metric

where *gateway* is the name or the IP address of a default router and *metric* is a preference indicator, which ranges from 1 (highest) to 15 (lowest).

```
#GatewayMetric
192.9.200.101
eng_gateway2
```



NOTE: Each entry for such a default router must have an IP address that belongs to the IP subnet of one of the interfaces configured for the filer.

How the filer replies to requests

The following list describes how the filer uses its interfaces to respond to different types of packets:

- NFS-over-UDP requests

  The filer does not use the conventional IP routing mechanisms to reply to NFS over UDP requests. The filer sends the response on the network interface on which the request was received, to the same address that generated the request. For example, the filer named toaster uses the interface toaster-e1 to send packets in response to NFS requests received on the interface toaster-e1.

  This way of handling NFS over UDP requests enables you to attach multiple interfaces of the filer to networks with the same IP subnetwork number while keeping NFS over UDP traffic isolated to the appropriate physical networks.

  Because of this scheme, it is possible that NFS over UDP responses might be returned through a different path than you might expect from an examination of the IP routing table using `netstat -r`. This scheme generally works well, although it can result in different routes than expected if your environment contains one-way routes. (For example, the IP packets might

not be routed as you intended if you configured the network so that the IP traffic from host1 to host2 is routed through router1 and the IP traffic from host2 to host1 is routed through router2.

- NFS over TCP, CIFS, and HTTP requests

  The filer tries to return NFS-over-TCP and HTTP traffic over the interface on which the traffic was received. However, there are exceptions. For example, if the filer experiences excessive queuing to a response or experiences a time-out followed by a retransmit, the filer routes the traffic by using conventional IP routing table lookups. If the filer has multiple interfaces attached to networks with the same IP network number, the filer uses the first interface it finds with that number to send the responses.



NOTE: If the NFS-over-UDP, NFS-over-TCP, CIFS, or HTTP traffic takes place on an ATM interface, the filer always uses conventional IP routing table lookups and routing for that traffic.

- IP-based traffic other than NFS and HTTP requests

  For other types of traffic (for example, traffic generated by `telnet`, `rsh`, and `ping`), the filer uses IP routing table lookups and routing. If the filer has multiple interfaces attached to networks with the same IP network number, the filer uses the first interface it finds with that number to send the responses.

Using ifconfig to configure an interface

The */etc/rc* file in the root volume contains `ifconfig` commands to configure network interfaces at system boot. You can also manually run `ifconfig` when the system is operating; the command syntax is as follows:

```
ifconfig interface [ address ] [ netmask mask ]
[ broadcast address ] [ mediatype type ]
[ mtusize size ] [ up | down ] [ trusted | untrusted ]
```

or

```
ifconfig -a
```

Usually, you use the `ifconfig` command for the following purposes:

- Changing the interface's IP address, network mask, or broadcast address. The following are examples of configuring an Ethernet interface on a filer:

```
ifconfig e1 292.9.299.6
ifconfig e2 netmask 255.255.0.0
ifconfig e3 broadcast 292.9.299.255
```

  The interface names on your system might be different, depending on your filer model. For information about interface naming conventions, see "Naming conventions for network interfaces" on page 22.

- Setting the media type on an Ethernet interface. The following examples show configuring an Ethernet interface on a filer:

```
ifconfig e1 mediatype tp
ifconfig ns1 mediatype thick
ifconfig ns2 mediatype thin
```

  The media types you can use depend on the type of Ethernet card. The possible types you can enter in the `ifconfig`

command are the same as those you can select when running `setup`. They are described in Table 4-3.

Table 4-3. Media types for Ethernet interfaces

Media type	Description
100tx	100Base-T
100tx-fd	100Base-T, full-duplex
tp	10Base-T
tp-fd	10Base-T, full-duplex
auto	10Base-T / 100Base-TX Ethernet using autonegotiation
1000fx	gigabit Ethernet
thick	thick Ethernet
thin	thin Ethernet

- Setting the maximum transmission unit (MTU). The following example shows setting the MTU for an FDDI interface on a NetApp F330:

  ```
  ifconfig f0 mtusize 1500
  ```

  By default, the MTU is 4352 on an FDDI interface, 1500 on an Ethernet interface, and 9188 on an ATM interface. Use a smaller MTU value for an interface if a bridge or router on the attached network cannot break large packets into fragments.

  To view the current setting of the MTU value, use the `netstat -i` command.

- Configuring the interface up or down. For example:

  ```
  ifconfig e1a up
  ifconfig f1 down
  ```

If you want changes made with `ifconfig` to remain in effect after a reboot, include the `ifconfig` commands in the */etc/rc* file.

To show the current configurations of all network interfaces, enter the following command:

```
ifconfig -a
```

To show the current configuration of a specific network interface, enter the following command:

```
ifconfig interface
```

For example, to show the current configuration of interface e0, enter the following command:

```
ifconfig e0
```

For more information about the `ifconfig` command, refer to the *na_ifconfig*(1) man page.

ATM PVCs

The filer dynamically assigns SVCs (Switched Virtual Circuits) when interoperating with ATM hosts and with switches that support the FORE IP/SPANS protocols. To interoperate with ATM hosts and with switches that do not support FORE IP/SPANS, you must use ATM PVCs. PVCs are static; for each destination, you must establish (attach the IP layer to) a PVC explicitly and delete (detach the IP layer from) the PVC explicitly.

For each destination that needs to establish a PVC with the filer, you must establish an outgoing and an incoming PVC in at least three places:

- at the filer
- at the destination ATM host
- on all interconnecting ATM switches

Establishing an outgoing PVC on the filer

To establish an outgoing PVC on the filer, use the following filer command:

`arp -s` *`hostname`* `pvc` *`iface vpi vci`* `[`*`aal`* `[`*`encap`* `[`*`addr`*`]]]`

where

hostname is the name or IP address of the remote host.

iface is the name of the ATM interface.

vpi is the VPI (virtual path identifier); this must be 0.

vci is the VCI (virtual channel identifier); this number must have the following properties:

- It must not be in use on your filer.
- It must be less than 1,024.
- It must obey the limits of the destination host and interconnecting devices.

aal is the ATM adaptation layer type. It must be 4 or 5, and should be the AAL type supported by the destination host (typically 5). The default is 5.

encap is the encapsulation type. The Specify one of the following encapsulation types:

  - null (no encapsulation; this is the default)
  - llc_routed (IEEE LLC encapsulation for routed PDUs)
  - llc_bridged_8023 (IEEE LLC encapsulation for Ethernet / 802.3 bridged PDUs)

It should also be the same as the encapsulation type used by the destination host.

If the encapsulation type is llc_bridged_8023, you must include *addr.*

addr is a six-byte colon-separated destination MAC address.

Establishing an incoming PVC on the filer

To establish an incoming PVC on the filer, use the following command at the filer:

`arp -l` *`pvc iface vpi vci`* [*`aal`* [*`encap`*]]

Use the variables as described in the previous section.

Establishing a PVC on the remote ATM host

Set up a PVC on the remote ATM host according to the documentation for that host.

Establishing a PVC on interconnecting ATM switches

On the interconnecting ATM switches, assign virtual channels corresponding to the VPI and VCI entries made on the filer and the remote ATM host.

Displaying information about a PVC

To display information about all PVCs (and other interfaces) on a host, use the following command:

```
arp hostname
```

where *hostname* is the name of a host.

To display information about all the current PVCs, use the following command:

```
arp -a
```

A display similar to the one in Figure 4-2 appears.

```
toaster> arp -a
aye (272.16.1.1) at 0:0:3b:80:36:88
bdellium (272.16.1.2) at 0:0:3b:80:36:d4
ctenophore (272.16.1.3) at 0:0:3b:80:36:d7
mnemonic (272.16.1.4) at 0:0:3b:80:2e:5b
oedipus (272.16.1.5) at 8:0:20:80:8d:47
psych (272.16.1.6) at 8:0:20:80:8d:d3
xenon (272.16.1.7) at 8:0:20:81:d9:47
yttrium(272.16.1.9) at 8:0:20:7a:65:19
Incoming ATM Connections:
iface=a5 switch.port=f21a2420.56 vpi.vci=0.114 aal=5
    encapsulation=NULL
iface=a5 switch.port=f21a2420.25 vpi.vci=0.113 aal=5
    encapsulation=NULL
```

Figure 4-2. Output from the arp -a command

Deleting a PVC

To delete an outgoing PVC entry, use the following command:

```
arp -d hostname
```

where

hostname is the name or IP address of the remote host.

To delete an incoming PVC for a remote host, use the following command:

```
arp -x iface vpi vci
```

where

iface is the name of the interface.

vpi and *vci* have the values of the VPI and VCI of the PVC to be deleted for the specified interface.

Chapter 5: Sharing Files between NFS and CIFS Users

This chapter describes how the filer works with NFS and CIFS clients simultaneously. Because these two types of clients interact with a file server differently, it is important for you to understand how the read and write operations performed by one type of client affect the operations performed by the other type of client. This chapter discusses the following topics:

- CIFS and NFS file locks
- how the filer manages file locks when both NFS and CIFS users try to read or write to the same file
- how the filer supports CIFS access to symbolic links
- how NFS and CIFS use the per-file read-only bit
- how the filer manages names of files used by both NFS and CIFS users
- how much time the filer takes to convert directories that were created by NFS clients to be used by CIFS clients

This chapter does not discuss the differences between procedures that take place on network clients. For example, the way in which users gain access to files and directories depends on the NFS and

CIFS protocols: NFS users *mount* the directories that were exported, and CIFS users *share* directories to which they were given access in the Access Control List. For more information about these procedures, refer to the documentation for the clients' operating systems.

This chapter also does not discuss filer administration procedures for individual file-sharing protocols. Procedures such as rebooting the filer, recovering files from snapshots, and backing up the filer are different depending on whether there are CIFS clients at your site. The procedures are described in other chapters of this guide.

File-locking interactions

This section describes in general what happens when a client program using one protocol tries to read or write a file that is currently used by a client program using a different protocol in an environment consisting of CIFS, NFS(PC), and NFS users. For details about locking in a particular protocol, consult the documentation for that protocol.

There are two kinds of NFS clients: UNIX-based NFS clients and NFS(PC) clients. The NFS(PC) clients and CIFS clients behave in the same general way.

NFS locks are advisory, while CIFS locks are mandatory. CIFS applications depend on locking to behave properly. Because NFS locks are only advisory, file-manipulation operations (such as `rm`, `rmdir`, and `mv`) by a UNIX-based NFS client on a file opened by a NFS(PC) application can cause the application to crash.

Reads by UNIX-based NFS clients always succeed.

A deny-write operation causes a file-manipulation operation (such as rm, rmdir, and mv) on CIFS-accessed file by a UNIX-based NFS client to fail.

Byte-range locks work on portions of a file. Byte-range operations other than reads by a UNIX-based NFS client fail if the attempted operation is forbidden by the lock. As is appropriate for NFS, a UNIX-based NFS application might be forbidden to access a byte-range that is locked by CIFS.



NOTE: There is one exception to the enforcement of locks set by CIFS clients on the filer. When the filer runs the dump command, it ignores the file lock set by a CIFS client that prevents read access to parts of a file or the entire file. Ignoring the “read” file lock allows the filer to back up all files.

Managing symbolic links for CIFS access

CIFS clients can follow symbolic links, which are created by NFS clients. A symbolic link is a special file that points to another file or directory. A symbolic link is, in some respects, like a shortcut in the Windows environment.

There are two kinds of symbolic links: absolute and relative.

- Absolute symbolic links begin with a slash (/) and are treated as a path relative to the root of the file system.
- Relative symbolic links begin with a character other than a slash (/) and are treated as a path relative to the parent directory of the symbolic link.

You can control CIFS access to symbolic links in three ways:

- Enabling or disabling symbolic links
- Redirecting absolute symbolic links
- Preventing or allowing the following of symbolic links that can refer to a directory higher in the same tree

Enabling symbolic links

When the symbolic links for CIFS feature is enabled (the default), if the object being accessed by a CIFS client is a symbolic link (whether absolute or relative), the filer follows the link under the following conditions:

- The ultimate target is in the same share as the symbolic link.
- A symbolic link encountered in any path component other than the final one is always followed.
- The final component of a symbolic link is followed only if the operation is to open an existing file.

- Other operations, such as deleting and renaming, result in deleting or renaming the symbolic link itself rather than the target of the symbolic link.

CIFS client applications often perform operations such as writing to a temporary file, renaming the original file to a backup name, then renaming the temporary file to the original name. Therefore, take care in using symbolic links whose ultimate target is a file (as opposed to a directory). If the original file were targeted directly by a symbolic link, this sequence of operations would have the result—unintended by the application—of the file being stored in the directory where the symbolic link was, and the renamed symbolic link pointing at the original file rather than to the updated file. For symbolic links to directories, this type of situation does not arise.

Because many PC applications work as described previously, if there are symbolic links that point to files, a PC could encounter such symbolic links. It is best to disable symbolic links for CIFS when there are symbolic links that point to files.

If you expect many files to be changed by applications that update files as described, you might want to disable symbolic links for CIFS.

You enable and disable symbolic links with the `cifs.symlinks.enable` option. The option is On by default.

To disable symbolic links for CIFS, use

```
options cifs.symlinks.enable off
```

To enable symbolic links for CIFS, use

```
options cifs.symlinks.enable on
```

Redirecting absolute symbolic links

In a UNIX environment, the NFS client interprets the file system location represented by an absolute symbolic link. The CIFS client cannot do this. In a CIFS environment, the filer enables you to redirect absolute symbolic links on the filer. For example, you might want to redirect symbolic links pointing at the directory /u/users/charlie to the directory */home/charlie* on the filer. You do so by specifying symbolic link redirection mappings in a text file named */etc/symlink.translations*.

The format of */etc/symlink.translations* is

```
Map link target
```

where both *link* and *target* are absolute symbolic link path names.

For example, the entry

```
Map /u/users/charlie/* /home/charlie/*
```

makes symbolic links pointing at the directory */u/users/charlie* point to the directory */home/charlie*.

Preventing symbolic link cycling

It is possible to create directory structures that are cyclic by creating a symbolic link that refers to a directory higher in the same tree (through use of a symbolic link having a "dot" or "dot-dot" component). Therefore, a simple recursive descent of the tree goes deeper and deeper until the maximum path length is reached. At that point an error is returned. So, for example, if you used Windows Explorer to search for files in such a cyclic directory, the same files show up repeatedly.

The `cifs.symlinks.cycleguard` option controls whether symbolic links that might include a directory higher in the same tree are followed.

To eliminate the possibility of cyclic directory structures, make sure that the `cifs.symlinks.cycleguard` option is On (it is On by default) with the following command:

```
options cifs.symlinks.cycleguard on
```

If you use symbolic links having dot or dot-dot components and want the filer to follow the links, set the `cifs.symlinks.cycleguard` option to Off with the following command:

```
options cifs.symlinks.cycleguard off
```

When you list the contents of a directory, symbolic links that are valid references to files or directories are listed as if the target of the symbolic link existed in the directory. If the symbolic link cannot be expanded, it still looks like a file in a directory listing; however, any attempt by an application to open the link results in an access error.

NFS and CIFS use of the read-only bit

The filer, along with MS-DOS and Windows, supports a per-file read-only bit that reflects whether a file is writable or read-only. This bit applies only to files and not to directories. NFS has no protocol operations that know about the per-file read-only bit. However, some software, when used both by NFS clients on UNIX systems and by CIFS clients on Windows systems, requires that the read-only bit reflects whether the file is writable.

The following list describes how NFS treats the read-only bit:

- Any file with the read-only bit turned on is treated, for all NFS operations, as if it had no write permission bits turned on.
- If a file has at least one write permission bit turned on and an NFS client turns off all write permission bits, the filer turns on the read-only bit for that file. (As described in the previous paragraph, a file with the read-only bit appears to an NFS client not to have any write permission bits turned on.)
- If a file has no write permission bits turned on and an NFS client turns on any write permission bit, the filer turns off the read-only bit for that file.
- If a file's read-only bit is turned on and an NFS client attempts to find out the permission bits for the file, the actual permission bits for the file are not sent to the NFS client; instead, the filer sends the permission bits to the NFS client with the write permission bits masked off.
- If a file's read-only bit is turned on and a CIFS client turns the read-only bit off, the filer turns on the owner's write permission bit for the file.
- Files with the read-only bit turned on are writable only by the super-user.

Whenever the read-only bit is turned on by a client (even if it was already on before the client did so), the filer tracks whether the client that turned on the bit was an NFS or CIFS client, as follows:

- If the bit was turned on by a CIFS client, renaming the file is not allowed. This is because file systems on MS-DOS and Windows systems do not allow renaming a file whose read-only bit has been set. NFS deletes follow NFS conventions. That is, deletes are allowed if the user has write permission in the parent directory.
- If the bit was turned on by an NFS client that turned off all write permission bits, removing or renaming the file is allowed if the user has sufficient permission to do so. This is because file systems on UNIX systems allow removing or renaming a file that has no write permission bits set.
- If the filer is using UNIX-style security, CIFS clients are also allowed to delete a file with the read-only bit set. This is required for compatibility with standard UNIX source control programs, such as RCS.
- If the filer is configured with PC-style security, the read-only bit is enforced

Naming files used by both NFS and CIFS

File naming conventions depend on both the network clients' operating systems and the file-sharing protocols. For example, file names are case-sensitive for clients running UNIX and are case-insensitive, but case-preserving, for clients running any Windows operating systems. This section discusses how different kinds of clients can obtain access to the same files despite different file naming conventions.

Maximum length of file names

On the filer, the maximum length of a file name is 255 characters for NFS clients and CIFS clients that support the PC's long file name format. Some CIFS clients, such as MS-DOS and Windows 3.*x* clients, support only file names in the 8.3 format (8 characters for the file name and 3 characters for the file name extension). In any directory that has access from a CIFS client, the filer creates and maintains two names: the original long name and an additional short name in 8.3 format. The filer generates the 8.3 name as follows:

1. It truncates the file name to six characters.
2. It appends a tilde (~) and a number or letter to the name. If it runs out of letters and numbers because there are too many similar names, it creates a unique file name that bears no relation to the original file name.
3. It truncates the file name extension to three characters.



NOTE: The number or letter appended to the short name ensures that the file name is unique. It is not for showing the order of file creation.

For example, if an NFS client creates a file named *specifications.html*, the short name created by the filer is *specif~0.htm*. If this short name already exists, the filer uses a different number at the end of the file name. For example, if the UNIX client creates another file named *specifications_new.html*, the short version of *specifications_new.html* is *specif~1.htm*.

The short names appear on clients that support only the 8.3 format. The short names are not visible to NFS clients. On Windows 95 and Windows NT clients, you can choose to display the short name or the long name by using File Properties.



NOTE: Under some circumstances, an application running on a client that uses names in 8.3 format can "lose" the file's original long-format name. This can occur as a consequence of the way an application saves a file that it has edited. Some applications rename the original file, then save the edited file as if it were newly created. The filer thus receives instructions to delete the original file and create a new one. When the client supports only 8.3 names, this new name no longer has an equivalent in long format.

Legal characters used in file names

The characters that you can use in file names depend on the client operating systems. Because restrictions on legal characters vary from one operating system to another, refer to the documentation for your client's operating system for more information about prohibited characters.

When you name a file to be shared by users on different operating systems, it helps to use only characters that are common to both. For example, if you use UNIX to create a file and use a colon (:) as its file name, an MS-DOS user sees the name displayed as ~0 because the colon is an illegal character in an MS-DOS file name.

Case-sensitivity in file names

Uppercase and lowercase characters are significant to NFS clients but not to CIFS clients. For example, if a file named *specifications* already exists, an NFS user can still save another file under the name *Specifications*, but a CIFS user is instructed by the application to choose another file name. This section describes how both NFS and CIFS users can use file names that differ only by case.

When a client creates a file name, the filer preserves the case. For example, if a CIFS client creates *Spec.txt*, the file name is displayed by both CIFS and NFS clients as *Spec.txt*. If an NFS user later creates a file named *spec.txt*, NFS and CIFS clients see the file names as follows:

- On NFS clients, one file is displayed as *Spec.txt* and the other is displayed as *spec.txt*. That is, the file names are displayed in the same way as they were created.
- On CIFS clients, even those that support long names, one file is displayed as *Spec.txt* and the other is displayed as *Spec~0.txt*.

Timing of directory conversion for CIFS access to NFS-created files

If you license both NFS and CIFS protocols, your filer supports access through either protocol to the same files. File-naming conventions differ between NFS and CIFS; although any CIFS-based file name is valid in NFS, some NFS names are not valid in CIFS. Also, CIFS maintains for each file certain attributes for which there are no NFS equivalents. A filer that runs CIFS maintains a directory in which NFS-style names are assigned equivalents in the 8.3 format.

If you add CIFS access to files that were created by NFS clients, the first time a directory is accessed through CIFS, the filer automatically converts it to the format required to support CIFS clients. "Accessed" here includes listing the contents of a directory, for example, with the File Manager, Explorer, or DOS `dir` command, or by accessing any files or directories contained in that directory.

Under some circumstances, the CIFS client might automatically look at the directory's subdirectories, thereby triggering conversion of the subdirectories as well. (For example, this happens if the user has set the Expand All option under the File Manager Tree menu.)

After a directory is converted, it does not have to be converted again.

Although this process is automatic, the initial conversion of a directory can take considerable time, especially if the directory contains a large number of files. For example, conversion of a directory of 60,000 files takes the filer about 35 minutes on a NetApp F540, and about 2 hours on a NetApp F330. It is important to take these conversion times into consideration when deciding when to convert the directories.

While a directory is being converted, the filer might not be able to perform any other file system operations until the conversion is complete.

If there are portions of the directory tree that will never have CIFS access, there is no need to convert them. However, any future CIFS access to an unconverted directory immediately triggers its conversion.

If you have a directory that contains more than 50,000 files, before triggering a conversion, use an NFS client to distribute files among a greater number of subdirectories. This speeds up the conversion process and avoids a possible crash.

Chapter 6: **NFS Administration**

This chapter discusses the following topics:

- exporting volumes or directories to NFS clients
- configuring a filer for (PC)NFS
- configuring a filer for WebNFS
- displaying NFS statistics

Exporting volumes or directories

You can export volumes or directories on the filer to individual clients or to subnets.

Exporting to individual clients

To export volumes or directories from the filer, list them in the filer */etc/exports* file and run the `exportfs` command. For example:

```
toaster> exportfs -a
```

This command exports all the directories listed in the */etc/exports* file. To stop exporting a directory, use the `exportfs -u` command.

The /etc/exports file

The */etc/exports* file lists all directories that the filer can export by the `exportfs` command. By default, the root volume (*/vol/vol0*) and the */vol/vol0/home* directory are exported to the administration host. The default *exports* file contains the following lines:

```
#Auto-generated by setup Mon Oct 27 14:15:36 PST 1997
/vol/vol0 -access=adminhost,root=adminhost
/vol/vol0/home -root=adminhost
```

In your */etc/exports* file, *adminhost* is the name of the administration host that you entered during system setup.

The following list explains the contents of the default *exports* file:

- Only the administration host can mount the / directory in the root volume and modify the files in that directory. This directory contains the filer configuration files, such as */etc/hosts* and */etc/exports*.
- Clients anywhere have full access to any files in the */home* directory, with the administration host having root permission.

NOTE: The default */etc/exports* file gives all clients read and write permissions to */home*. If only users of particular groups should have permissions to read and modify files in */home*, use the `chmod` command to change modes for */home*. You can also use the `-rw` or `-access` option in */etc/exports* to limit write privilege to listed hosts. The `-rw` option lists the hosts that can modify the exported directories; hosts not listed by the `-rw` option have read privilege only. The `-access` option restricts mount access to listed hosts. Refer to the *na_exports*(1) man page for more information about options in the *exports* file.

You can specify up to 256 host names for each `-rw` and `-root` option. There is no limit on the number of host names for the `-access` option, provided that each line in the *exports* file has no more than 1,024 characters. If a line exceeds 1,024 characters due to a long list of clients, use netgroups in the list. For more information about netgroups, refer to "The /etc/netgroup file" on page 122.

In the *exports* file, you must list volumes separately if you want clients to be able to mount them. That is, you cannot export all volumes on the filer collectively by specifying */vol* as the exported directory.

Be sure that the filer can resolve all host names specified in the *exports* file. Refer to "Name resolution" on page 84 for more information about host-name-to-IP-address resolution.

NOTE: Although you *can* specify in */etc/exports* which clients can mount files, the filer (and NFS) doesn't offer a mechanism for you to specify a list of clients that cannot mount the filer.

In many UNIX implementations, it is illegal to export a directory that has an exported ancestor in the same file system. The filer has no such restriction. For example, it is legal to export both the */vol/vol0* volume and the */vol/vol0/home* directories.

The filer uses the longest matching prefix in determining permissions. In the preceding example, a client mounting */vol/vol0/home/user1* gets permissions for */vol/vol0/home* because */vol/vol0/home* is the longest matching prefix. A client mounting */vol/vol0* gets `-access=adminhost,root=adminhost` permissions. Because of this, it makes little sense to give a client greater permissions at a higher level in the file system, as shown in the following sample *exports* file:

```
/vol/vol0
/vol/vol0/home -access=bashful:dopey:sleepy
```

In this sample file, only the three clients specified in the second line can mount */vol/vol0/home* directly, but any client can access the */vol/vol0/home* subtree by mounting */vol/vol0*. This allows a breach in system security and should be avoided.

Instead of naming individual clients in the `-access` option of */etc/exports* or the `exportfs` command, you can specify a group of clients (a network group) defined in the */etc/netgroup* file. See the next section on network groups.

After editing */etc/exports*, either execute `exportfs -a` or `reboot` to make the new configuration take effect.



NOTE: If you are deleting entries in the new*/etc/exports* file, you need to run `exportfs -au` first and then `exportfs -a`; otherwise, the previously loaded *exports* file remains in effect.

The /etc/netgroup file

The filer */etc/netgroup* file defines groups of clients that the filer uses for checking access permission while processing a mount request. Each line of this file defines a group and has the following format:

```
groupname member-list
```

Each element in `member-list` is another group name or a triplet of the following form:

(*hostname*, *username*, *domainname*)

An element can be blank, but the commas must be present.

Because only the group name is used in the filer *exports* file or `exportfs` command, you don't have to specify the *username* and *domainname* fields. Changes made to the netgroup file take effect immediately.

In */etc/netgroup*, the maximum number of characters per line is 4,096. Following is an example:

```
trusted-hosts (adminhost,,)
untrusted-hosts (bashful,,) (dopey,,) (sleepy,,)
all-hosts trusted-hosts untrusted-hosts
```

With this */etc/netgroup* file, you can write entries in */etc/exports* as follows:

```
/vol/vol0 -access=trusted-hosts,root=adminhost
/vol/vol0/home -access=all-hosts,root=adminhost
```

Do not use netgroup names in the lists for the `root=` and `rw=` options in the */etc/exports* file.

If your filer is not configured as an NIS client, the network groups on the filer are not linked with NIS. An existing NIS network group must be copied over to the filer */vol/vol0/etc/netgroup* before it can be used by `exportfs`. You can modify the Makefile of the NIS master to copy the NIS master's */etc/netgroup* to the filer when it is changed. For example, you can put the following code in the NIS Makefile section for `netgroup.time`. Substitute the name of each filer in the "for" list in place of *toaster1*, *toaster2*, and so on.

```
@mntdir=/tmp/nac_etc_mnt_$$$$;\
if [ ! -d $$mntdir ]; then rm -f $$mntdir; mkdir $$mntdir; fi;\
for filer in toaster1 toaster2 toaster3 ; do \
mount $$filer:/vol/vol0/etc $$mntdir;\
mv $$mntdir/netgroup $$mntdir/netgroup.bak;\
cp /etc/netgroup $$mntdir/netgroup;\
umount $$mntdir;\
done;\
rmdir $$mntdir
```

Exporting to subnets

You can export a directory to clients on a subnet rather than to individual clients. You can include netgroups in the exported paths unless the path is exported with the `-rw` or `root` options.

You export a directory to a subnet as you do to an individual client, but you specify a subnet address rather than a full IP address in an export option. The valid export options in the */etc/exports* file for exporting to a subnet are as follows:

```
-o ro=subnet_address[:subnet_address]...
-o rw=subnet_address[:subnet_address]...
-o root=subnet_address[:subnet_address]...
```

The subnet address is a dotted IP subnet address and a mask written in the following format:

```
dotted_ip/num_bits
```

where *dotted_ip* can be an IP address (“a.b.c.d”) or an IP subnet (a for a class A network, a.b for a class B network, or a.b.c for a class C network). The size of the subnet is specified by the number of leading bits of the netmask, *num_bits*.

Following are some examples of exporting subnets:

- To export */vol/vol0/home* on the filer for root access to a client named "tooting" and all addresses of the form "123.45.67.x" with a netmask 255.255.255.0 (24 leading bits), enter this command:

  exportfs -o root=tooting:123.45.67.0/24 /vol/vol0/home

- To export */vol/vol0/home* for read and write access to all addresses of the form "123.45.x.y" with a 16-bit netmask (255.255.0.0), enter this command:

  exportfs -o rw=123.45.0.0/16 /vol/vol0/home

The following `exportfs` commands are equivalent. They export */vol/vol0/home* to a client named "host1," the specified subnet, and a client named "host2."

```
exportfs -o rw=host1:123.45.67.8/24:host2 /vol/vol0/home
exportfs -o rw=host1:123.45.67/24:host2 /vol/vol0/home
```

Effects of volume name change on /etc/exports

If you use the `vol rename` command to change the name of a volume, the change is not automatically incorporated into */etc/exports*. If the old volume name is referenced in the *exports* file, the filer prints an error message the next time it executes the `exportfs -a` command. The message states that the volume under the old name is not available and cannot be exported. However, the in-memory information about exported volumes is updated after each `vol rename` command. That is, if you enter the `exportfs` command without any arguments to display currently exported volumes, you see the new volume name in the output.

Update *etc/exports* immediately after renaming a volume. With an inaccurate */etc/exports*, the filer cannot export the volume under the old name after it reboots. Also, clients that mounted the volume before the reboot receive the "Stale NFS file handle" message when trying to access that volume.

Configuring a filer for (PC)NFS

The pcnfsd authentication daemon enables you to use (PC)NFS to access a filer. (PC)NFS refers to NFS client implementations for DOS, Windows, Windows NT, OS/2, and Macintosh operating systems. Using (PC)NFS, such systems can mount a directory or volume from the filer after the user provides the UNIX user name and password for authentication. Without the pcnfsd daemon running on the filer, you must set up another host to provide the authentication.

The filer supports (PC)NFS version 2 except for printer support. With (PC)NFS version 2, clients can use the filer for authentication, including obtaining group permissions (GIDs) and group names through the */etc/group* file or the corresponding group.byname NIS map.

To use (PC)NFS to access a filer, the client system must run (PC)NFS and must be configured to mount the filer. Also, the pcnfsd daemon can run only if NFS is licensed for your filer. Otherwise, the following warning messages appear:

```
pcnfsd is NOT running
Cannot use pcnfsd without nfs.
(Use the “nfs on” command to enable nfs.)
```

Starting the (PC)NFS server

On the filer, perform these steps to start the (PC)NFS server:

1. Enable the pcnfsd daemon, entering the options command:

   ```
   toaster> options pcnfsd.enable on
   ```

   To preserve this option setting after filer reboots, place this command in the filer */etc/rc* file.

2. Verify that each user of (PC)NFS has an entry in your UNIX */etc/passwd* file (and */etc/shadow* file, if applicable).

3. If the filer has not been configured to use NIS for the password database, copy */etc/passwd* and */etc/shadow* (if applicable) to the filer. For example, on the UNIX system where */etc/passwd* and */etc/shadow* reside, enter the following commands:

```
% cp /etc/passwd /nfs/toaster/etc/passwd
% cp /etc/shadow /nfs/toaster/etc/shadow
```

In this example, the mount point for the filer root volume is */nfs/toaster*.

To make sure these files on the filer are always up to date, modify the Makefile for the NIS master to copy them to the filer. For example, put the following code at the end of the NIS Makefile section for `passwd.time`, substituting the name of each filer in the "for" list in place of "toaster1," "toaster2," and so on. Add the line copying the */etc/shadow* file only if you are using a shadow password file.

```
@mntdir=/tmp/nac_etc_mnt_$$$$;\
if [ ! -d $$mntdir ]; then rm -f $$mntdir; mkdir $$mntdir; fi;\
for filer in toaster1 toaster2 toaster3 ; do \
mount $$filer:/vol/vol0/etc $$mntdir;\
mv $$mntdir/hosts $$mntdir/hosts.bak;\
cp /etc/passwd $$mntdir/passwd;\
cp /etc/shadow $$mntdir/shadow;\
umount $$mntdir;\
done;\
rmdir $$mntdir
```

If you are not using NIS, you can establish a `cron` job to update the filer.

The filer is now ready to serve files to NFS clients using (PC)NFS. Each user on the client system can mount the file system from the filer after entering a correct user name and password.

Default umask

You can set a default umask for clients with the following `options` command:

```
options pcnfsd.umask nnn
```

where *nnn* is the umask.

The default umask is 022, which means that each user can read and write his or her own files, but group members and others can only read the files.

Configuring a filer for WebNFS

The filer can use NFS rather than HTTP to respond to file transfer requests made through a browser that supports WebNFS clients. The headers of WebNFS requests have URLs that start with "nfs://".

With WebNFS, the filer can transfer files much faster than with HTTP because WebNFS can transfer several files (including graphics files) with only one TCP connection. HTTP (in version 1.1) requires a separate connection for each file. The filer does not need an HTTP license to respond to WebNFS requests.

Restricting access

WebNFS access does not use the `mount` command to enable access to a subtree, and does not consider UID/GID mapping. Requests are restricted as follows:

- Requests are honored only for files in subtrees that have been exported.
- If a subtree has been exported with the `-o access` option, files in that subtree are not available through WebNFS.

For information about the `exportfs` command, consult the *na_exportfs*(1) man page.

Turning on WebNFS

To turn on WebNFS, set the `nfs.webnfs.enable` option to the `options` command to On, as follows:

```
options nfs.webnfs.enable on
```

To turn WebNFS off, set the option to off, as follows:

```
options nfs.webnfs.enable off
```

Specifying and setting the root directory

You can specify a public directory (known as the root directory) for WebNFS access. All NFS lookups are done relative to the root directory. If WebNFS is enabled on the filer, all WebNFS clients can access files and directories under the root directory.

Specifying the root directory

To specify the root directory, use the `options nfs.webnfs.rootdir` command with the following syntax:

```
options nfs.webnfs.rootdir directory
```

Setting the root directory

To make the WebNFS root directory take effect, enter the following `options` command:

```
options nfs.webnfs.rootdirset TRUE
```

Changing the root directory

To change the WebNFS root directory, follow these steps:

1. Use the the following command syntax:

   ```
   options nfs.webnfs.rootdir newdir
   ```

2. For the new root directory to go into effect, enter the following command:

   ```
   options nfs.webnfs.rootdirset TRUE
   ```

If you use the `vol rename` command to change the name of the volume in which the WebNFS root directory resides, remember to use the preceding procedure to specify the new name of the root directory.

Disabling the root directory

To disable the root directory, enter the following command:

```
options nfs.webnfs.rootdirset FALSE
```

Displaying NFS statistics

The `nfsstat` command displays statistics on NFS and RPC for the filer. You can use the output of this command to find performance bottlenecks or inefficiencies in your NFS setup.



NOTE: A full description of the meaning of NFS statistics is outside the scope of this guide. A good source of information about this topic is *Managing NFS and NIS* by Hal Stern, O'Reilly & Associates, Inc.

The syntax of the `nfsstat` command is as follows:

```
nfsstat [ interval ]
nfsstat -h [ ip_address | host_name ]
nfsstat -l
nfsstat -z
```

The `nfsstat` command is explained in detail later in this section.

With no arguments, `nfsstat` displays statistical information since the last time you rebooted the filer.

If you specify an interval (in seconds), `nfsstat` continually displays summary information. Following is an example of `nfsstat` output.

```
toaster> nfsstat
Server rpc:
TCP:
calls      badcalls   nullrecv   badlen     xdrcall
0          0          0          0          0

UDP:
calls      badcalls   nullrecv   badlen     xdrcall
24         0          0          0          0

Server nfs:
calls      badcalls
24         0

Server nfs V2: (24 calls)
null     getattr   setattr  root      lookup  readlink  read
0 0%     5 21%     0 0%     0 0%      16 67%  0 0%      0 0%
wrcache  write     create   remove    rename  link      symlink
0 0%     0 0%      0 0%     0 0%      0 0%    0 0%      0 0%
mkdir    rmdir     readdir  statfs
0 0%     0 0%      3 13%    0 0%

Server nfs V3: (0 calls)
null     getattr   setattr  root      lookup  readlink  read
0 0%     0 0%      0 0%     0 0%      0 0%    0 0%      0 0%
write    create    mkdir    symlink    mknod  remove    rmdir
0 0%     0 0%      0 0%     0 0%       0 0%   0 0%      0 0%
rename   link      readdir  readdir+  fsstat  fsinfo    pathconf
0 0%     0 0%      0 0%      0 0%      0 0%   0 0%      0 0%
commit
0 0%
```

To enable per-client NFS statistics, enter the following command:

options nfs.per_client_stats.enable on

If you enable per-client NFS statistics, use the following options to display them:

- To display a list of the clients whose statistics have been collected, along with the total NFS calls for that client, use the

`nfsstat -l` command. Following is an example of `nfsstat -l` output.

```
toaster> nfsstat -l
172.17.25.13  sherlock   NFSOPS =     2943506 (90%)
172.17.25.16  watson     NFSOPS =     3553686 ( 2%)
172.17.25.18  hudson     NFSOPS =     2738083 ( 1%)
172.17.230.7  conan      NFSOPS =     673247l ( 3%)
172.17.230.8  baker      NFSOPS =   202614527 ( 1%)
172.17.230.9  moriarty   NFSOPS =     1006881 ( 0%)
175.17.230.10 doyle      NFSOPS =        1185 ( 0%)
```



NOTE: If you want per-client statistics, enable per-client statistics collection as soon as possible after you start the filer or reset the counters with `nfsstat -z`. Otherwise, `nfsstat -l` reports incorrectly low percentages and displays statistics that include clients that have generated RPC calls but no NFS calls to the client.

- To display details for one client only, use `nfsstat -h` with the client IP address or host name as an additional argument:

  `nfsstat -h` *`hostname`*

Following is an example of `nfsstat -h` output:

```
toaster> nfsstat -h eng_host
Client: 172.17.25.3 (eng_host)  -------------------------------
Server rpc:
calls   badcalls nullrecv badlen  xdrcall
33374   0        0        0       0

Server nfs:
calls   badcalls
33345   0

Server nfs V2:
null    getattr  setattr  root     lookup     readlink  read
0 0%    8410 25% 19 0%    0 0%     13687 41%  42 0%     489 22%
wrcache write    create   remove   rename     link      symlink
0 0%    3225 10% 12 0%    7 0%     9 0%       0 0%      0 0%
mkdir   rmdir    readdir  statfs
0 0%    0 0%     416 1%   29 0%

Server nfs V3:
null    getattr  setattr  lookup   access     readlink  read
0 0%    0 0%     0 0%     0 0%     0 0%       0 0%      0 0%
write   create   mkdir    symlink  mknod      remove    rmdir
0 0%    0 0%     0 0%     0 0%     0 0%       0 0%      0 0%
rename  link     readdir  readdir+ fsstat     fsinfo    pathconf
0 0%    0 0%     0 0%     0 0%     0 0%       0 0%      0 0%
commit
0 0%
```

To reset counters, enter the following command:

`nfsstat -z`

For more information about `nfsstat`, see the *na_nfsstat*(1) man page.



NOTE: The information displayed by the `nfsstat` command (including per-client NFS statistics) is also available through SNMP if you use the Network Appliance custom MIB. For more information about the custom MIB, see "New Network Appliance custom MIB" on page 82.

Chapter 7: **CIFS Administration**

This chapter describes how to use filer commands for these tasks:

- adding clients to the filer
- using `cifs` commands with `rsh`
- enabling guest access
- creating a home directory for a CIFS user
- managing shares
- understanding the limits of open files, sessions, and shares
- choosing the method for deciding access rights
- managing access rights
- using CIFS oplocks
- using NetBIOS scope IDs
- displaying CIFS statistics
- displaying CIFS session information
- stopping and restarting CIFS service
- reconfiguring the filer for CIFS

This chapter does not discuss procedures that take place on the clients. Also, it does not describe how a user joins a workgroup or domain. For information about these topics, refer to the manuals for your client operating systems or books about PC networking.

Adding clients to the filer

You can add CIFS users to the filer at any time. The following steps outline the procedure for adding a CIFS user:

1. For users to be authenticated, follow one of these steps:
   - If you use a domain controller for user authentication, create an account for the CIFS user on the domain controller that is used for authenticating users. For more information about this step, refer to the manual for the system used as a domain controller.
   - If you use a UNIX password database for user authentication, enter the users' information into the NIS password and group maps. If you do not use NIS, create entries for the user in the filer */etc/passwd* and */etc/group* files.
2. If necessary, modify the ACL (Access Control List) for each share to be used by the user. For more information about managing access rights, refer to "Managing access rights for users or groups" on page 158.



NOTE: How you manage file access rights depends on whether you specified during `setup` that the filer uses PC style or UNIX style to determine users' access rights. Refer to "Choosing the method for deciding access rights" on page 151 to see whether you need to manage the UNIX-style permissions for directories created for the new user.

Using cifs commands with rsh

You can use `cifs` commands with `rsh` (remote shell). For example, on the administration host, you can set access rights for a user as follows:

```
adminhost% rsh -l root -n toaster cifs access home jsmith
r-x
```

In this example, the filer name is “toaster,” the user is “jsmith,” and the share is “home.” The user has read and execute rights to the directory on the filer that has been defined as the *home* share.

Because you can use `cifs` commands through `rsh`, you can automate the task of defining access rights for multiple CIFS filers with similar user information. For example, you can create a script containing `cifs` commands to enter user information for each filer at your site.

Enabling guest access

You can enable users who do not have an account on the filer to have access to the filer through CIFS in two ways:

- To give access to users who are *not* in a trusted domain, use a guest account.
- To give access to users who *are* in a trusted domain, use a generic user account.

Guest account

If you use a domain controller for authentication, guests are users who are not in trusted domains.

If you use a UNIX password database for authentication, guests are users who do not have an entry in the database.

Use the following option:

`options cifs.guest_account` *`account_name`*

where *account_name* is the name of the guest account, usually `guest` or `nobody` (a preconfigured account in the */etc/passwd* file), to provide a guest account. To disable guest access, use a blank as *account_name*.

Set the guest account to the name of an account in the UNIX password database (typically `guest`, which is mapped to the UNIX account `nobody` with the same access rights as the user `everyone`).

Generic account

If you are using a domain controller, you can allow users who are in trusted domains but who do not have a password on the filer or in the UNIX password database to connect to the filer. To create such an account, use the following option:

```
options cifs.generic_account account_name
```

where *account_name* is the name of the generic account. All users in the generic account have the same rights.

Creating a home directory for a CIFS user

To create a home directory for a CIFS user, use the NetApp Permissions Tool. The NetApp Permissions Tool and its documentation are available from the NetApp On the Web (NOW) web site at *http://now.netapp.com*.

After creating a home directory, you can offer the directory as a share to a user, as described in "Creating a "home" share for each user" on page 146.

You can also use the `cifs mkdir` command. The syntax for this command is as follows:

`cifs mkdir` *`share user path`*

where *share* is a share, *user* is the login of a user, and *path* is a path relative to the root directory of share.

The `cifs mkdir` command enables you to use a PC to create for a UNIX user a directory having the appropriate UNIX-style permissions, and at the same time to make the user the directory owner. The directory has *user* as its owner and *user*'s primary group as its group owner.

Managing shares

This section describes how to create a share, display information about shares, and delete a share. By default, two shares are created: the hidden share *C$* and *HOME*. The *C$* share corresponds to */vol/vol0* and the *HOME* share corresponds to */vol/vol0/home*.

Creating a share

If you want a directory to be shared by CIFS clients, use the `cifs shares` command to define a new share. Following is the syntax for `cifs shares`:

`cifs shares -add` *`sharename path description`*

The following list describes the parameters in the command:

- *sharename* is the name of the share, which is used by CIFS users to obtain access to the directory on the filer. The name must be a valid 8.3 name.

  If *sharename* already exists, the `cifs shares -add` command overwrites the old information.



CAUTION: Do not create shares whose names end with a dollar sign ($); doing so might cause conflicts with reserved names. In particular, do not create shares called C$ or ADMIN$. C$ is a reserved share name and ADMIN$ is an illegal share name.

- *path* is the complete path name of the shared directory on the filer.



CAUTION: Because the `cifs shares` command is case-sensitive, be sure that you use the appropriate case when entering the path name.

Separators used in the path name must be forward slashes (/).

- *description* is a string describing the purpose of the share. It is required by the CIFS protocol and is displayed in the share list

in Network Neighborhood on the client. If the description contains spaces, it must be enclosed in single quotation marks.

The following example creates a new share:

```
cifs shares -add library /vol/vol0/home/lib 'New file library'
```

This example creates the *library* share and defines it as */home/lib* in the filer's root volume. With the appropriate access rights, CIFS users can gain access to the */vol/vol0/home/lib* directory, which is displayed as the *library* share on their computers. For more information about setting access rights, refer to "Managing access rights for users or groups" on page 158.



NOTE: Do not rename a share if a file in the share is currently opened by a client application. For example, you can use the `cifs shares` command to rename the *library* share to *new_lib* only if no files in */vol/vol0/home/lib* are currently opened by any CIFS clients.

Changing the description of a share

To change the description of a share, enter the following command:

```
cifs shares -change sharename description
```

The following list describes the parameters in the command:

- *sharename* is the name of the share whose description you want to change.
- *description* is a string describing the new purpose of the share. If the description contains spaces, it must be enclosed in single quotation marks.

You can change the description of a share even if the share is in use. The change takes effect immediately.

Displaying information about shares

To display information about the shares that you defined, enter the following command:

`cifs shares [`*`sharename`*`]`

The following command displays information only for the share *sharename*:

`cifs shares `*`sharename`*

The following command displays information about all shares:

`cifs shares`

The following sample output is from `cifs shares`:

```
Name          Mount Point      description
HOME          /vol/vol0/home    Default Share
                   rw- laura
                   rw- bob
                   rwx jane
C$           /vol/vol0/         Remote Administration
                   rwx george
```

Share permissions are shown by triplets consisting of `rwx` or a dash replacing one or more of the letters, as follows:

- `r` indicates read permission granted, while a dash in the first position indicates read permission denied.
- `w` indicates write permission granted, while a dash in the second position indicates write permission denied.
- `x` indicates execute permission granted, while a dash in the third position indicates execute permission denied.



NOTE: Delete permission is included in the write permission. Browse permission is included in the execute permission.

Deleting a share

You can delete a share only if client applications have no open files in the share. To delete a share, use the following command syntax:

```
cifs shares -delete sharename
```

Following is an example:

```
toaster> cifs shares -delete library
```

The preceding example deletes the share created by the `cifs shares -add` command described in "Creating a share" on page 143.

Creating a "home" share for each user

You can create a share that contains home directories of registered CIFS users and a directory for guest (assuming the guest access option is set to the PC convention "guest"). For example, if there are users called user1 and user2, the share contains directories for user1, user2, and guest. When user1 connects to the filer and asks for its list of shares, the display shows an offer called user1, but not user2 or guest (or any other individual user).

A user accesses his or her home directory in the same way as any other share. That is, the user can open the share with Network Neighborhood, by mapping a drive, or by using a UNC name. The UNC name is simply

```
\\filer\username
```

User name longer than the maximum-length share name

Because share names are truncated to 12 characters, the home directory name might show a truncated version of the user's account name. For example, consider the 13-letter name "administrator". From the filer point of view, the home directory for "administrator" has a name that exactly matches the account name "administrator", but use of that directory is offered under the truncated share name "administrato".

(If there were users “administrator” and “administrator1”, each would see an offer for his or her own home directory, but both would see it as an offer to share the name “administrato”. Thus, each gets the correct directory, even though the share names appear the same.)

Creating a share containing user home directories

The following procedure creates a share that automatically contains home directories for CIFS users:

1. Enable the home directory option by entering the following command:

   `options cifs.home_dir` *`homedirpath`*

   This creates a share named “CIFS.HOMEDIR.” This share automatically contains CIFS user home directories. The *homedirpath* parameter in the `options` command is the path name corresponding to the share. For example, if the directory containing the home directories is */vol/vol1/homes*, enter the following command:

   `options cifs.home_dir /vol/vol1/homes`

   When you display the filer’s list of shares, the */vol/vol1/homes* directory appears in the shares list, which is mapped to the share named “CIFS.HOMEDIR,” as shown in the following example:

```
toaster> cifs shares
Name            Mount Point       Description
C$              /vol/vol0/         Remote Administration
                ** priv access only **
CIFS.HOMEDIR    /vol/vol1/homes    Home Directories
                rwx everyone
```

2. Set up a directory for each user.

Within the directory specified by the `cifs.home_dir` option, create a directory for each user. Each directory name must exactly match the user's login name. Also, make each user the directory's owner.

Using the `cifs.home_dir` option to create home directories is useful only if users do not need to read or write other users' home directories. If they must access other users' home directories, create an additional share that maps to the same path name as the CIFS.HOMEDIR share. Then assign each user the appropriate access permissions to other users' home directories. The following example shows how to create an additional share, assign user access rights to the share, and display the share information:

```
toaster> cifs shares -add enghomes /vol/vol1/homes "Readable home
directories"
toaster> cifs access enghomes -g engineering r-x
toaster> cifs shares
Name           Mount Point          Description
enghome        /vol/vol1/homes      Readable home directories
               r-x techpubs{g}
C$            /vol/vol0             Remote Administration
               ** priv access only **
HOME          /vol/vol0/home        Default Share
               ** priv access only **
CIFS.HOMEDIR  /vol/vol1/homes       Home Directories
               rwx everyone
```

In the preceding example, users in the group "engineering" can read all home directories in */vol/vol1/homes*, which corresponds to the share defined by the `options cifs.home_dir` command. However, they can only write to their own home directories, which reside in the *CIFS.HOMEDIR* share.

Effects of volume name change on CIFS shares

If you use the `vol rename` command to change the name of a volume that contains a share, the filer automatically offers the share to users in the renamed volume at the next reboot. The new volume name is reflected in the `cifs shares` command.

Limits on CIFS open files, sessions, and shares

The filer is subject to limits on files access through CIFS, as summarized in Table 7-1.

Table 7-1. CIFS limits by filer memory configuration

	64 MB	128 MB	192 MB	256 MB
Maximum number of users	128	523	1,024	2,048
Maximum number of open files	2,560	10,240	20,480	40,960
Maximum number of locked files	2,048	4,096	6,144	8,192
Maximum number of shares	1,000	1,000	1,000	1,000

Choosing the method for deciding access rights

When a CIFS user tries to obtain access to a file on the filer, the filer uses one of two methods for determining a user's file access rights: PC style and UNIX style. The two methods are offered because share-level ACLs do not work properly with the UNIX-style method but do work properly with the PC-style method. You choose the method when you set up CIFS for the first time. If you want to change the method, run the `cifs setup` command, which prompts you for the method.

This section explains how you select the method based on the mix of users at your site. It also provides examples comparing the advantages of PC style and UNIX style, depending on the needs of the users. The method affects only CIFS users.

PC style

If the filer uses PC style, when a CIFS user tries to obtain access to a file, the filer checks the share-level ACL before checking the UNIX-style permissions. (The filer gives each file UNIX-style permissions upon file creation.) The following example illustrates how the filer combines the information from the share's ACL and the file's UNIX-style permissions to decide the users' file access rights.

In this example, the shared directory is */vol/vol0/home/hr-users*; the group "hr" consists of "chris" and "laura." The ACL for */vol/vol0/home/hr-users* is as follows:

```
HR-USERS    /vol/vol0/home/hr-users    Sample HR Share
                 r-x hr
                 r-x barbara
                 rwx laura
                 rwx chris
```

The ACL includes the following information:

- All members of "hr" have read and execute rights. The group "hr" is defined in the filer */etc/group* file.
- In addition to having read and execute rights, laura and chris can write to */vol/vol0/home/hr-users*.
- Barbara does not belong to "hr," but she has read and execute access.
- Users outside of "hr" and excluded from the ACL have no access.

When a user attempts to write to a file within the share, the filer first checks whether the user is accorded the right to write by the share ACL. If a user doesn't have the right to write according to the ACL, the filer rejects the attempt to write to */vol/vol0/home/hr-users*. Otherwise, the filer checks to see whether the file's UNIX-style permission allows the user to write. (File permissions were inherited from the file's parent directory when the file was created.) For example, barbara's attempt to write fails because the ACL doesn't give her the write privilege. However, laura has the right to write, so her attempt to write depends on the UNIX-style permissions of the file, as follows:

- If laura is the file owner, whether she can write depends on the file's write permission that is applicable to the file owner.
- If laura is not the file owner, whether she can write depends on the file's write permission that is applicable to the group. (The group permissions apply to everyone other than the file owner. The filer disregards the GID.)

PC style offers these advantages:

- ACLs work.
- You do not need a UNIX client to manage file permissions. You can leave the file permissions at their default settings, and use

share-level ACLs to control the access rights of CIFS users. By default, the permissions of the */home* directory are as follows:

```
drwxrwx---
```

If your site requires private directories for CIFS users, use the NetApp Permissions Tool (available from NOW) on a PC to modify permissions for the private directories even when you use PC style. For example, if */vol/vol0/home/hr-users* contains chris' */vol/vol0/home/hr-users/chris_home* private directory, to prevent users other than chris from reading or writing to it, change the permissions for *chris_home* to the following:

```
drwx------
```

UNIX style

Using UNIX style, when a CIFS user tries to obtain access to a file, the filer checks the UNIX-style permissions before checking the share-level ACL. The filer gives each file UNIX-style permissions upon file creation.

When UNIX style is used, the GID of a file is significant. That is, the filer compares the GID of the file with the GIDs of a user to determine whenever the user belongs to a particular group. The filer applies the group permissions only to members of the group as specified in the UNIX group database.

If you are using UNIX style permissions, in general it is best to set the ACL to everyone and rely on UNIX permissions to control access.

Use the following guidelines to determine whether the UNIX style is suitable for your site:

- If the CIFS users also use NFS to mount directories and are familiar with UNIX-style file permissions, the users can use UNIX clients to manage the permissions of the files they create. For example, if chris creates a file named */vol/vol0/home/hr-users/policy*, she can set the UNIX permissions

to prevent users other than herself from reading and writing to the file.

- If you are satisfied with the way UNIX systems define groups, use the UNIX style to minimize the work required for managing access rights. For example, if you want all members of the "hr" group to have the same rights, in the ACL, assign read, write, and execute rights to the group "hr." In this way, you can use UNIX permissions for each file exclusively for managing access rights; you need to add entries to ACL only if you want to limit the access rights for particular users of the "hr" group.

Changing UNIX-style permissions for files in a CIFS share

In addition to using the NetApp Permissions Tool, you can use the `cifs chmod` command to set UNIX-style permissions of a designated file or directory within a share. The syntax is as follows:

`CIFS chmod` *`share mode path`*

The arguments are as follows:

share The name of a CIFS share.

mode The access mode to be applied to the file or directory, as described in the tables that follow. You can specify the mode in either absolute format or symbolic format.

path The path, relative to the root directory of the specified share, to the file or directory whose access is to be set.

Absolute Mode. A mode specified in absolute format is an octal number. The full range of options is represented by a number having four octal digits, but for most purposes a 3-digit number suffices (with an implied leading 0).

The significance of digits in the various positions is the same as in UNIX systems, as shown in Table 7-2.

Table 7-2. Absolute mode values

Value	Position Execution mode[a]	Owner	Group	Others
4	Set-user-ID-on-execution	Read	Read	Read
2	Set-group-ID-on-execution	Write	Write	Write
1	Sticky bit	Execute / Search	Execute / Search	Execute / Search

a. These bits might be significant when a UNIX system executes the file, but they do not affect access rights.

For example, the mode 644 gives the owner read and write permission and gives group members and others read permission.

The mode 755 gives the owner read, write, and execute permissions, and group members and others read and execute but not write permission.

The mode 0 denies every permission to all.

Symbolic Mode. A mode specified in symbolic format is a comma-separated list of expressions. Each expression is formed from up to three components, as shown in Table 7-3.

Table 7-3. Symbolic mode components

Who		Operation		Permission	
u	User	+	Add	r	Read
g	Group	-	Remove	w	Write
o	Others	=	Set	x	Execute / search
a	All (that is, ugo)			s	Set-user-ID-on-execution and Set-group-ID-on-execution
				t	Sticky

For example, the mode expression `go-w` denies write permission to group and others. The mode expression `u=rwx,go=rx` gives the user (that is, the file owner) permission to read, write, and execute, and gives members of the user's group and others permission to read and execute (but not write). It is thus equivalent to 755.

An expression that uses = but omits the permission clears all permissions. For example, the mode expression `go=` turns off all access for group and others, and is equivalent to `go-rwxst`.

An expression that does not indicate *who* applies to all. For example, the mode `+x` adds execute permission to all, and is equivalent to `a+x`.

Changing the ownership of files in a share

The `cifs chown` command changes the owner of a file or directory. The syntax is as follows:

`cifs chown` *`share owner path`*

The arguments are as follows:

share	The name of a CIFS share.
owner	The user name of the new owner.
path	The path, relative to the root directory of the specified share, to the file or directory whose access is to be set.

Managing access rights for users or groups

This section describes how to assign access rights and remove an entry in a share's ACL. It also describes how to display the rights of users and groups.

Assigning access rights

After you create a share, define the users' or groups' access rights to the share. To assign access rights, use the `cifs access` command with the following syntax:

```
cifs access share [ -g ] user|group rights
```

The `-g` flag specifies that the access rights are defined for a group.

The rights are defined as `r`, `w`, and `x`, which mean read, write, and execute, respectively. To deny a right to a user, use a minus sign (-). Following are some examples:

```
cifs access library -g engineering rwx
cifs access library joed r--
```

The group "everyone" is reserved. When it is used in an ACL, the group "everyone" means every CIFS user. For example, to give every CIFS user read, write, and execute rights to the *library* share, enter the following command:

```
cifs access library -g everyone rwx
```

Removing an entry from a share's ACL

If a group or a user no longer exists and you want to remove the corresponding entry in an ACL, use the following command syntax:

```
cifs access -delete share user | group
```

Following are some examples:

```
cifs access -delete library engineering
cifs access -delete library joed
```

Displaying access rights to a file

To display access rights to a file, use the `cifs rights` command. It has the following syntax:

```
cifs rights share user path
```

where *share* is a share, *user* is the login of a user, and *path* is the path of a file relative to the root directory of share.

This displays the rights that the user has for the file with the path name *path* relative to the root of the share *share*. Following is an example of the `cifs rights` command:

```
toaster> cifs rights C$ hawleyr /etc/rc
PC style Access Control is being used
user is a member of group eng
access rights for share LINK = r-x | r-x | r-x
access rights for share ENG-USERS = rwx | rwx | rwx
access rights for share HOME = r-x | r-x | r-x
access rights for share C$ = rwx | rwx | rwx
full path = /vol/vol0/etc/rc
File Unix permissions user= Administrator(root)
  group= well
  file perm =rwx | rwx | ---
  PC rights RWX
  UNIX rights ---
```

Using CIFS oplocks

Oplocks (opportunistic locks) enable the redirector on a client in certain file-sharing scenarios to perform client-side caching of read-ahead, write-behind, and lock information. A client can, therefore, work with a file (read or write it) without regularly reminding the server that it needs access to the file in question, thus improving performance by reducing network traffic.

Under some circumstances, if a process has an exclusive oplock on a file and a second process attempts to open the file, the first process must relinquish the oplock and access to the file. The redirector must then invalidate cached data and flush writes and locks, resulting in possible loss of data that was to be written. Oplocks are, therefore, best used in an environment where most operations are reads rather than writes.

Any application that has write-cached data can lose that data under the following set of circumstances:

- It has an exclusive oplock on the file.
- It is told to either break that oplock or close the file.
- During the process of flushing the write cache, the network or target system generates an error.

The cache itself does not have any error handling—the applications do. When the application makes a write to cache, the write is always completed. If the cache in turn makes a write to the target system over a network, it must assume that the write is completed because if it does not, the data is lost.

CIFS oplocks on the filer are on by default.

You might turn CIFS oplocks off under either of the following circumstances:

- You are using a database application whose documentation recommends that oplocks be turned off.

- You are handling critical data. That is, you have a good network but you cannot afford data loss.

Otherwise, you leave CIFS oplocks on.

You turn off CIFS oplocks with the following `options` command:

```
options cifs.oplocks.enable off
```

You turn on CIFS oplocks with the following `options` command:

```
options cifs.oplocks.enable on
```

For additional information about oplocks, consult the Microsoft Knowledge Base at *http://www.microsoft.com/kb*.



NOTE: You can turn CIFS oplocks on or off at individual clients. Turning CIFS oplocks on at the filer does not override any client-specific settings. Turning CIFS oplocks off at the filer disables all oplocks to or from the filer.

Using NetBIOS scope IDs

The NetBIOS scope ID feature provides a naming service for the NetBIOS-over-TCP/IP (NBT) module. The NetBIOS scope ID feature is not compatible with DNS. The scope ID is a case-sensitive character string that is added to the end of a NetBIOS name.



NOTE: Microsoft recommends that you not use NetBIOS scope ID if you are not currently using it.

Do not enable scope ID for the filer unless you are already using it for other machines because you can unintentionally isolate the filer from clients that do not have the same NetBIOS scope ID.

If you are not using DNS, NetBIOS scope ID enables you to limit traffic to only those machines with the same NetBIOS scope ID.

You can use this feature to isolate a group of computers on a network so they can communicate only with other computers with the identical NetBIOS Scope ID, or to build virtual subnetworks (groups of machines that see only each other). By default, the scope ID is a null string, but if the filer is to run in a NetBIOS scope other than the default one, its scope ID must be set to the ID of that scope. The scope ID can be changed only when CIFS is not running.

To enable NetBIOS scope ID, use the following command snytax:

`options cifs.scope.id` *`scopeID`*

The default is null, which means that no scope ID is used.

To disable NetBIOS scope ID, enter the following command:

`options cifs.scope.id`

Displaying CIFS statistics

The `cifs stat` command displays statistics about CIFS operations that take place on your filer.

You can use the `cifs stat` command in two forms:

- If a time interval is specified, the command displays statistics at the specified intervals. For example, the following command displays statistics every second:

  `cifs stat 1`

  Following is an example of the output of `cifs stat 1`.

```
toaster> cifs stat 1
  GetAttr    Read   Write    Lock   Open/Cl   Direct   Other
      175     142      54      70       254      643      50
      232      76     123      44       321      154      17
      152     120      34     111        12      435      76
```

- If no time interval is specified, the `cifs stat` command displays CIFS statistics that have accumulated since the last reboot. The `cifs stat` display is for diagnostic purposes.

Displaying CIFS session information

To display information about connected users, the number of shares and open files for each user, enter the following command:

```
cifs sessions
```

To limit the information to a particular user or client name, specify the user name or client name in the command. Following is an example:

```
cifs sessions joed
```

Following is a sample output from the `cifs sessions` command without a user name.

```
Server Registers as 'TOASTER' in group NT-DOMAIN
WINS Server: 172.20.0.4
PC style Access Control is being used
Using domain controller NT-DOMAIN-PDC for authentication
=======================================
PC (user)           #shares       #files
SMITHPC (jsmith)    1             1
PETERSPC (mpeters)  2             3
```

Stopping CIFS service

You can stop CIFS service by using the `cifs terminate` command. Always use this command before you reboot or turn off the filer. The `halt` command automatically invokes the `cifs terminate` command.

Although the `reboot` command also stops CIFS service, the `cifs terminate` command provides a time delay during which users can save their open files before the disconnection. Changes that have not been saved to disk are lost if the CIFS client has an open file when it is disconnected from the filer. The default time delay is five minutes. You can specify the time delay in minutes by using the `-t` option. For example, to terminate CIFS service after 10 minutes, enter the following command:

```
toaster> cifs terminate -t 10
```

The `cifs terminate` command also displays an alert message on CIFS clients that warns them of the pending shutdown. Windows 95 and Windows for Workgroup clients must have the WinPopup program configured before they can display the alert message.

If you want to terminate CIFS service for a particular client, specify the name of the computer in the command. For example, the following command terminates CIFS service to a computer named PETERSPC after 10 minutes:

```
toaster> cifs terminate PETERSPC -t 10
```

Following is an example of the `cifs terminate` command.

```
toaster> cifs terminate
There are currently 35 CIFS users that have 37 open files
Disconnecting while files are open may cause data loss!!
How many minutes should I wait? [5]:
minutes left 4
minutes left 3
minutes left 2
minutes left 1
CIFS shutting down
```

If you want to cancel the `cifs terminate` command, press Ctrl-C before the end of the countdown.



NOTE: The `cifs terminate` command disables CIFS service only between the time you enter the command and the next reboot. After each reboot, if your filer is licensed and configured to run CIFS, the filer automatically starts CIFS service. If you want to prevent CIFS from restarting after a reboot, remove */etc/cifsconfig.txt* from the filer.

Restarting CIFS service

To restart CIFS service, use the `cifs restart` command. For example:

```
toaster> cifs restart
CIFS server is registering...
CIFS server is running.
```

Reconfiguring the filer for CIFS

If you want to reconfigure the filer for CIFS service, for example, if you want to change the authentication method from PDC to UNIX password database, follow these steps:

1. Enter the `cifs terminate` command to stop CIFS service.

2. Enter the `cifs setup` command to reconfigure CIFS service. For more information about the prompts displayed in `cifs setup`, refer to the *Quick Start Guide*.

   After you enter `cifs setup`, the filer copies the file */etc/cifsconfig.txt* to */etc/cifsconfig.bak*. If the information entered during `cifs setup` is valid, a new */etc/cifsconfig.txt* file is created, and the filer automatically restarts CIFS using the new configuration information. If you enter information that is invalid, for example, if you mistyped a domain name during `cifs setup`, the filer restarts CIFS by reverting to the previous configuration. The contents of */etc/cifsconfig.txt* are restored from the */etc/cifsconfig.bak* backup file.

Chapter 8: **HTTP Administration**

This chapter describes how to perform the following tasks:

- starting HTTP service
- protecting web pages with passwords
- using the HTTP virtual firewall
- enabling virtual hosting
- specifying MIME Content-Type values
- translating URLs
- displaying HTTP connection information
- displaying HTTP statistics



NOTE: You can use the filer as an HTTP server only if you have purchased the license for HTTP. Without the license, you can use an HTTP client (web browser) only to display the filer's man pages and to use FilerView.

Starting HTTP service

To start HTTP service on your filer, follow these steps:

1. Enable the `httpd` daemon by entering the following command:

   `options httpd.enable on`

2. Use the following command syntax to specify the root directory that contains the files and directories to be read by HTTP clients:

   `options httpd.rootdir` *`directory`*

   For example, if the root directory is */vol/vol0/home/users/pages*, enter the following command:

   `options httpd.rootdir /vol/vol0/home/users/pages`

3. If you want to limit the size of the log file */etc/log/httpd.log* to other than the default of 2,147,483,647 bytes (2 GB minus 1 byte), use the following command:

   `options httpd.log.max_file_size` *`bytes`*

4. Make a copy of */etc/httpd.mimetypes.sample* and name the copy */etc/httpd.mimetypes*. If the */etc/httpd.mimetypes* file is missing, the HTTP client uses the information in */etc/httpd.mimetypes.sample*.



NOTE: If you want these options to remain active after rebooting, you must add them to the */etc/rc* file.

The procedure for starting HTTP service is now complete, and clients can display text files under the root directory by using a web browser. If the filer will transfer files other than text files, for example, image files and audio files, follow the instructions in "Specifying MIME Content-Type values" on page 178 to configure your filer so that the appropriate MIME Content-Type header is sent with each file transferred.

To test the filer's HTTP service, follow these steps:

1. Create an HTML file in the root directory for HTTP. For example, create a file named *myfile.html* in the HTTP root directory, which is */vol/vol0/home/users/pages*, assuming that the HTML root directory is */vol/vol0/home/users/pages*.

2. Start a web browser on a client and specify the URL of the HTML file in the browser.

   For example, if your filer is *toaster* and the root directory for HTTP is */vol/vol0/home/users/pages*, enter this URL:

   ```
   http://toaster/myfile.html
   ```

The path component of the URL is a path name relative to the HTTP root. Do not specify the complete path name to the file in the URL.



NOTE: If the URL names a directory, for example, *http://toaster/home/pages*, the filer automatically tries to transfer the *index.html* file within the directory. If *index.html* does not exist, the filer returns "Error 404. No such file or directory."

Protecting web pages with passwords

You can restrict access to a specified directory so that only specified users or groups have access to it.

Password protection involves three configuration files:

- */etc/httpd.access*
- */etc/httpd.passwd*
- */etc/httpd.group*

The /etc/httpd.access file

The */etc/httpd.access* file contains directives that govern authentication for each directory. The filer supports the following directives:

- directory
- AuthName
- require user
- require group

These directives are compatible with the Apache web server directives, but the filer ignores all other directives.

The directory directive

This directive specifies a directory tree to be protected and encloses all other directives. The syntax of the `directory` directive is as follows:

```
<Directory directory_name>
directive ...
</Directory>
```

The AuthName directive

This directive specifies a "realm," which is an alias for the directory and appears instead of the directory name in the browser's password dialog box when a user tries to access the directory. Whatever follows `AuthName` is the name of the realm. The name of the realm can contain spaces. The syntax of the `AuthName` directive is as follows:

```
AuthName realm name
```

The require user directive

This directive specifies the users who can access the directory. The syntax of the `require user` directive is as follows:

```
require user user_id [, user_id, ... ]
```

The require group directive

This directive specifies the groups that can access the directory. The syntax of the `require group` directive is as follows:

```
require group group_id [, group_id, ... ]
```

The /etc/httpd.passwd file

The */etc/httpd.passwd* file contains *user_id* and encrypted-password pairs. The pairs have the following format:

```
user_id:encrypted_passwd
```

The pairs are copied in from a machine on which the user has a password.

The /etc/httpd.group file

The */etc/httpd.group* file contains a *group_id* and a list of *user_ids* in that group in the following format:

```
group_id: user_id [user_id  ....]
```

The lists are copied in from a machine that has a similar list.

Web page protection examples

The following */etc/httpd.access* file restricts access to */vol/vol0/home/htdocs/private/spec* to only user bob:

```
<Directory /vol/vol0/home/htdocs/private/spec>
AuthName polard Private Stuff
<Limit GET>
require user bob
</Limit GET>
</Directory>
```

<Limit GET> and </Limit GET> are not supported, but the file might have been imported from an Apache or NCA server. To be used on a filer, the file does not need to be edited to remove the Limit GET directive; the filer ignores the directive and all other directives not mentioned in this chapter.

The following sample procedure restricts user access to a particular directory:

1. Enter the following lines in the */etc/httpd.access* file:

   ```
   <Directory /vol/vol0/home/htdocs/private/specs>
   AuthName Social commentary
   require group engineering
   </Directory>
   ```

2. Enter the following line in */etc/httpd.group*:

   ```
   engineering: bob larry nancy rose
   ```

The */vol/vol0/home/htdocs/private/specs* directory is now accessible only to the group "engineering," which consists of the following user IDs:

bob, larry, nancy, and rose

Using the HTTP virtual firewall

The HTTP virtual firewall feature helps you maintain security on your filer. You restrict HTTP requests by marking the subnetwork interface over which they arrive as "untrusted." An untrusted interface provides only HTTP access to your filer on a read-only basis.

Mark an interface untrusted if it meets all the following conditions:

- You know you are going to service HTTP requests over that interface.
- You don't want to allow requests through protocols other than HTTP.
- You want to restrict access to the filer through that interface to read-only access.

By default, a subnetwork interface is trusted.

Mark an interface as untrusted or trusted by setting an option to the `ifconfig` command. Following are examples of the command:

- To mark the f0 interface as untrusted, enter the following command:

  `ifconfig f0 untrusted`

- To mark the f0 interface as trusted, enter the following command:

  `ifconfig f0 trusted`

Using virtual hosting

Virtual hosting enables a filer to respond to requests directed to more than one IP address through a single physical interface. This means that a filer with only one physical interface can host several IP addresses. Virtual hosting enables, for example, an Internet provider to host several web sites but have only one physical interface.

An HTTP server can use the destination IP address of an incoming HTTP request to find the directory that contains the HTTP pages belonging to the virtual host.

To enable virtual hosting, direct requests by putting entries in the */etc/httpd.hostprefixes* file and map virtual host addresses to the virtual host interface with the `ifconfig` command.

Using the /etc/httpd.hostprefixes file

Each line in the */etc/httpd.hostprefixes* file has the following format:

`prefix` [*`host-name-or-address ...`*]

The *prefix* field specifies a subdirectory in the HTTP root directory, which is defined by the `options httpd.rootdir` command.

The *host-name-or-address* field is an HTTP host name or an IP address. You can have more than one of each.

For example, the line

```
/customer www.customer.com 192.225.37.102
```

means that an HTTP request that comes for the interface with address 192.225.37.102, or with an HTTP 1.1 Host: header specifying *www.customer.com*, is directed to */customer*, and the requestor cannot get a file outside the */customer* directory.

If the HTTP server receives an HTTP request that is destined for one of its virtual host IP addresses, in this example 192.225.37.102, the destination IP address is used to select the virtual host root directory from the */etc/httpd.hostprefixes* file.

Mapping virtual host addresses

To map virtual host addresses to the virtual host interface, use the `ifconfig` command as follows:

- Add a new IP virtual host address mapping with the following command:

  `ifconfig vh alias` *`address`*

  where *`address`* is an IP address.

  The use of the `vh` interface indicates to the system that you are adding a virtual host address rather than adding an IP alias address to a network interface.

- Delete virtual host addresses with the following command:

  `ifconfig vh -alias` *`address`*



NOTE: If you need to create a virtual subnet with many contiguous addresses, the IP address can be a subnet address.

Specifying MIME Content-Type values

You can configure the filer to send the appropriate MIME (Multipurpose Internet Mail Extensions) Content-Type header in each response to a get request. The header shows the MIME Content-Type value of the file, which tells the browser on the client how to interpret the file. For example, if the MIME Content-Type value shows that the file being transferred is an image file and the client is configured properly, the browser can render the image by using a graphics program.

The filer determines the MIME Content-Type value of a file by mapping the file name suffix, for example, *.gif*, *.html*, or *.mpg*, according to information in */etc/httpd.mimetypes*.



NOTE: On a Windows 95 or Windows NT 4.0 client, the */etc/httpd.mimetypes.sample* file name is not displayed in its entirety. By default, the Explorer displays the file name as */etc/httpd.mimetypes*. If you are using Windows, from the Explorer View menu, select Options, then the View tab and, in the dialog box, make sure that there is no check mark in the box next to "Hide MS-DOS file extensions for file types that are registered."

To modify MIME Content-Type mappings or to add MIME Content-Types, edit the entries in */etc/httpd.mimetypes*. Entries are in the following form:

```
# An optional comment.
suffix   Content-Type
```

Lines preceded by the # sign are comments. Following are sample entries:

```
# My clients' browsers can now use
# PICT graphics files.
pct        image/pict
pict       image/pict
```

According to these entries, files whose names ended with *.pct* or *.pict* are mapped to the MIME Content-Type value of "image/pict." The file name suffix is case-insensitive. The first field in the Content-Type value describes the general type of data contained in the file; the second field is the data subtype, which shows the specific format in which the data is stored. If the browser on the client is configured to start a graphics program as a helper application, the user can view a file named *file.pict* as a graphics file on the client.

For more information about MIME, refer to RFC 1521.

Translating URLs

You can specify that the filer's response to an HTTP request be dependent on the URL. For example, you can configure the filer to redirect a particular request to a specific directory, or to prevent access to a particular directory that is specified in the URL. How the filer maps its responses to URLs is defined in a configuration file named */etc/httpd.translations*.

Each entry in */etc/httpd.translations* contains up to three fields in the following form:

```
rule template result
```

The `rule` field defines the response of the filer to a request. The `template` field specifies a component of a URL. The meaning of the `result` field depends on the rule, as described in the following section.

Translation rules supported by the filer

This section explains the meanings of rules. It also describes the format required for each type of entry in */etc/httpd.translations*.

The map rule

The map rule specifies that if a component of a URL matches the template, the request is mapped to another directory on the same host as defined in the `result` field. For example, the following */etc/httpd.translations* entry causes any requests to a URL containing the */image-bin* directory to be mapped to the */usr/local/http/images* directory:

```
Map /image-bin/* /usr/local/http/images/*
```

The redirect rule

The redirect rule specifies that if a component of a URL matches the template, the request is redirected to the URL defined in the `result` field. For example, if */etc/httpd.translations* contains the following entry

```
Redirect /cgi-bin/* http://cgi-host/*
```

the filer redirects CGI requests to another HTTP server named "cgi-host." This is essential for calls to *cgi-bin* because the filer does not execute them.



NOTE: The result field for the redirect rule must be specified as a complete URL beginning with *http://* and the host name.

The pass rule

The pass rule specifies that if a component of a URL matches the template, the filer accepts the request, processes the request as is, and disregards other rules. For example, if */etc/httpd.translations* contains the following entry

```
Pass /image-bin/*
```

the filer processes the request for any URL containing */image-bin* as is, even though there is another rule specified as follows:

```
Map /image-bin/* /usr/local/http/images/*
```

If the pass rule includes the `result` field, the filer accepts the request, processes the request by using the URL defined in the `result` field, and disregards other rules.

The fail rule

The fail rule specifies that if a component of a URL matches the template, the filer denies access to that component and disregards other rules. For example, if */etc/httpd.translations* contains the following entry

```
Fail /usr/forbidden/*
```

the filer does not provide access to the */usr/forbidden* directory.

How the filer processes rules

The filer processes the rules defined by */etc/httpd.translations* in the order they are listed, and applies the rule if the URL matches the template. However, the filer stops processing other rules after it applies a pass or fail rule.

In the `template` or `result` field of a */etc/httpd.translations* entry, you can use asterisks (*) as wildcard characters, as follows:

- In the `template` field, the wildcard character matches zero or more characters, including the slash (/) character.
- In the `result` field, the wildcard character represents the text expanded from the match in the `template` field. Include the wildcard character in the `result` field only if you used a wildcard character in the `template` field.
- If you use multiple wildcard characters, the first one in the `result` field corresponds to the first one in the `template` field, the second one in the `result` field corresponds to the second one in the template field, and so on.

Following is an example showing how a wildcard character is used:

```
# Redirect all cgi requests to my cgi server
Redirect /cgi-bin/*    http://cgi-host/cgi-bin/*
```

This redirect rule specifies that all CGI requests are redirected to another host named “cgi-host.” If the filer receives the following requests

```
http://toaster/cgi-bin/displayfares
http://toaster/cgi-bin/displaydates
```

the filer expands the wildcard character to “displayfares” and “displaydates” and redirects the requests to the host named “cgi-host.” To the client, the results of these requests are the same as the results of the following requests:

```
http://cgi-host/cgi-bin/displayfares
http://cgi-host/cgi-bin/displaydates
```

Displaying HTTP connection information

Read the */etc/log/httpd.log* file if you are interested in the following types of information for each HTTP connection:

- IP address of HTTP client.
- Name of authorized users (if the requested page is protected) making requests. The names are in the */etc/httpd.passwd* file. If the page is not protected, dashes appear instead of a name.
- Time of connection in `dd/mm/yy:hh:mm:ss` format (gmt is used).
- Request line from connecting host (for example, `get /netapp.html`).
- STATUS code returned by the server (as defined in the HTTP 1.0 specifications).
- TOTAL bytes sent in response by the filer, not including the MIME header.

Following is an example of the */etc/log/httpd.log* file:

```
192.9.77.2  - - [26/Aug/1996:16:45:50] "GET /top.html" 200 1189
192.9.77.2  - - [26/Aug/1996:16:45:50] "GET /header.html" 200 531
192.7.15.6  - - [26/Aug/1996:16:45:51] "GET /logo.gif" 200 1763
198.9.200.2 - - [26/Aug/1996:16:45:57] "GET /task/top.html" 200 334
192.9.20.5  authuser [26/Aug/1996:16:45:57] "GET /task/head.html"
200 519
```

Displaying HTTP statistics

The `httpstat` command displays statistics about HTTP operations on the filer. The `httpstat` command displays four types of statistics:

HTTP gets	Successful requests for files.
HTTP badcalls	Requests for nonexistent files.
open conn.	Number of HTTP connections currently open.
peak conn.	The largest number of simultaneous HTTP connections since the filer was booted or since the `-z` option was used.

The syntax for the `httpstat` command is as follows:

```
httpstat [ -t|-z ][ interval ]
```

Without any arguments, `httpstat` displays HTTP statistics accumulated since the last reboot or since the last time the `-z` argument was used. The `-z` argument resets both the gets and badcalls counters. The `-t` argument displays statistics since the last filer reboot.

You can specify the interval, in seconds, at which the filer displays the statistics. Following is an example of `httpstat`:

```
toaster> httpstat
HTTPD statistics:
gets         badcalls         open conn.   peak conn.
451          11                5            17
```

Chapter 9: **Snapshots**

A snapshot is a read-only copy of the entire file system—it reflects the state of the file system at the time the snapshot was created. Any client of a filer can access snapshots to recover old versions of files, for example, files that were accidentally changed or deleted. The snapshot feature enables users to restore their own files without your help, because files in snapshots can be viewed and copied by those who have permission to do so with the original files.

Snapshots also simplify tape backup. The filer `dump` command automatically creates a snapshot of the active file system, if necessary, before backing up the data to tape. However, it is not necessary if you are backing up an existing snapshot. Because a snapshot is a read-only copy of the file system, it does not change even when files in the active file system are changing. As a result, `dump` can make a safe and consistent backup without requiring you to take the filer off-line.

The filer uses a copy-on-write technique to create snapshots quickly without consuming any disk space. Snapshots begin to consume extra space only as blocks in the active file system are modified and written to new locations on disk. For more information about the copy-on-write technique used by snapshots, refer to "How snapshots work" on page 194.

The filer creates and deletes snapshots automatically at preset intervals. You can also create and delete snapshots manually. Each volume on the filer can have up to 20 different snapshots at one time.

Snapshot files carry the same permissions and inode numbers as the original files, keeping the integrity of the security system intact. Inodes are data structures that hold information about files on the filer. There is an inode for each file and a file is uniquely identified by the file system on which it resides and its inode number on that system.



NOTE: The inode number for a file in a snapshot is the same as the inode number for the corresponding file in the active file system. As a result, some programs on UNIX clients consider the two files to be the same. For example, if you use the GNU `diff` program to compare the two files, it does not find any differences between them. To make sure that the two files have different inode numbers before the comparison, copy one of the files to another name.

Snapshot commands

The commands related to snapshots are listed in Table 9-1. If the volume name is omitted in any of these commands, the command applies to the root volume.

Table 9-1. Commands for snapshots

Command	Meaning
`snap list`[a] *`volume_name`*	Lists all available snapshots.
`snap create` *`volume_name snapshot_name`*	Creates a snapshot with a specified name.
`snap delete` *`volume_name snapshot_name`*	Deletes a specified snapshot.
`snap rename` *`volume_name from to`*	Renames a snapshot.
`snap reserve` *`volume_name`*	Reserves a percentage of the disk space for snapshots.
`snap sched` *`volume_name`*	Schedules automatic snapshots.

a. Some of the information generated by this command is available through SNMP using the Network Appliance custom MIB. The custom MIB is described in Chapter 4, "Routine Filer Administration."

Automatic snapshot creation

The filer uses the snap sched command to create snapshots automatically and to keep them on-line for a predetermined amount of time.

There are three types of automatic snapshots:

- Weekly—The filer creates these every Sunday at midnight. Weekly snapshots are called *weekly.n*, where *n* is an integer. *weekly.0* is the most recent weekly snapshot, and *weekly.1* is the next most recent weekly snapshot. When the filer creates a weekly snapshot, the value of *n* is adjusted for all weekly snapshots. The higher the value of *n*, the older the snapshot.

- Nightly—The filer creates these every midnight except when a weekly snapshot is scheduled to occur at the same time. If the number of weekly snapshots is nonzero and it's the day of the week that weekly snapshots occur, no nightly snapshot is created. Nightly snapshots are called *nightly.n*, where *n* is an integer. *nightly.0* is the most recent nightly snapshot, and *nightly.1* is the next most recent nightly snapshot. When the filer creates a nightly snapshot, the value of *n* is adjusted for all nightly snapshots. The higher the value of *n*, the older the snapshot.

- Hourly—The filer creates these on the hour at specified hours, except at midnight, if a nightly or weekly snapshot is scheduled to occur at the same time. This occurs either if the number of nightly snapshots in the schedule is nonzero, or if the number of weekly snapshots in the schedule is nonzero and it's the day of the week that weekly snapshots occur. Hourly snapshots are called *hourly.n*, where *n* is an integer. *hourly.0* is the most recent hourly snapshot, and *hourly.1* is the next most recent hourly snapshot. When the filer creates an hourly snapshot, the value of *n* is adjusted for all hourly snapshots. The higher the value of *n*, the older the snapshot.

Figure 9-1 shows a sample snap sched command.

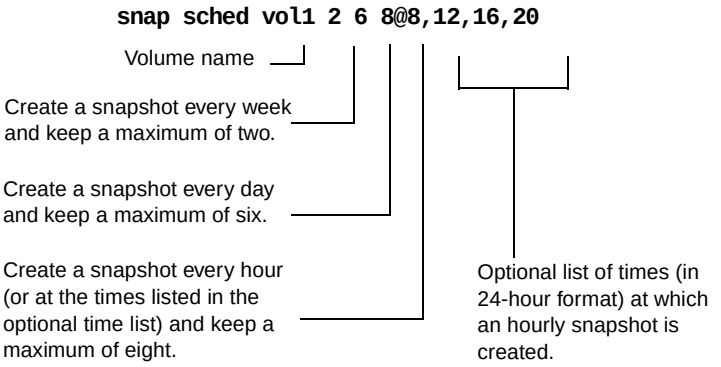


Figure 9-1. Example of the snap sched command

In the snap sched command, the first argument after the volume name indicates how many weekly snapshots to keep, the second argument indicates how many nightly snapshots to keep, and the third argument indicates how many hourly snapshots to keep. A zero in any of the three positions disables snapshots for that interval. The argument for hourly snapshots can include an optional list of numbers indicating the hours at which the filer creates the snapshots in 24-hour time. If the argument is omitted, the filer creates an hourly snapshot every hour.

The default snapshot schedule is

```
snap sched volume_name 0 2 6@8,12,16,20
```

Following is an example of the snap sched command:

```
snap sched volume_name 2 6 8@8,12,16,20
```

The following list describes the snapshots created by the example:

- weekly snapshots, and keeps the two most recent
- daily snapshots, and keeps the six most recent

- hourly snapshots at 8:00 a.m, noon, 4:00 p.m., and 8:00 p.m., and keeps the eight most recent

The following list shows the snapshots that are created by this snapshot schedule in 1998 (when January 11 is a Sunday).

```
% ls -lu .snapshot
total 64
drwxrwsrwx  2 root 4096 Jan 14 12:00 hourly.0
drwxrwsrwx  2 root 4096 Jan 14 08:00 hourly.1
drwxrwsrwx  2 root 4096 Jan 13 20:00 hourly.2
drwxrwsrwx  2 root 4096 Jan 13 16:00 hourly.3
drwxrwsrwx  2 root 4096 Jan 13 12:00 hourly.4
drwxrwsrwx  2 root 4096 Jan 13 08:00 hourly.5
drwxrwsrwx  2 root 4096 Jan 12 20:00 hourly.6
drwxrwsrwx  2 root 4096 Jan 12 16:00 hourly.7
drwxrwsrwx  2 root 4096 Jan 14 00:00 nightly.0
drwxrwsrwx  2 root 4096 Jan 13 00:00 nightly.1
drwxrwsrwx  2 root 4096 Jan 12 00:00 nightly.2
drwxrwsrwx  2 root 4096 Jan 10 00:00 nightly.3
drwxrwsrwx  2 root 4096 Jan 09 00:00 nightly.4
drwxrwsrwx  2 root 4096 Jan 08 00:00 nightly.5
drwxrwsrwx  2 root 4096 Jan 11 00:00 weekly.0
drwxrwsrwx  2 root 4096 Jan 04 00:00 weekly.1
```

This schedule keeps the eight most recent hourly snapshots (created at 8 a.m., noon, 4 p.m., and 8 p.m. every day), the six most recent daily snapshots, and the two most recent weekly snapshots. Whenever the filer creates a new snapshot of a particular type, it deletes the oldest one and renames the existing ones. On the hour, for example, the filer deletes *hourly.7*, renames *hourly.0* to *hourly.1*, and so on. The nightly snapshot schedule jumps from January 12 to January 10 because there is a weekly snapshot on January 11.



NOTE: On a UNIX client, if you use `ls -l` instead of `ls -lu` to list the snapshot creation times, the times are not necessarily all different. The times listed by `ls -l` reflect the modification times of the directory at the time of each snapshot, and are not related to the times at which the snapshots are created.

The snap sched command is persistent across reboots. There is no need to put the command in the */etc/rc* file.

You can create snapshots at predefined times instead of using the hourly, daily, and weekly schedules. For example, if you want to create two snapshots for the volume named "vol1" each week, you can set up a cron job on the administration host to run twice each week at an appropriate time to execute the following snap create command:

```
rsh toaster snap create vol1 filename
```

The snap create command does not accept a snapshot name containing a slash (/).

How snapshots work

When the filer creates a snapshot, it doesn't copy disk blocks; instead, it identifies all blocks in the file system as belonging to the snapshot as well as to the active file system. Consider a particular file named *foo* in a newly created snapshot. Because the snapshot was just created, the snapshot version of *foo* has the same contents as the version in the active file system. The same blocks on disk store both versions, so the snapshot version of *foo* consumes no disk space.

Later, if *foo* is deleted, the blocks holding the data for *foo* are no longer part of the active file system, but they are still part of the snapshot. Therefore, deleting *foo* from the active file system does not free any disk space. Figure 9-2 illustrates how disk space is used before and after *foo* is removed.

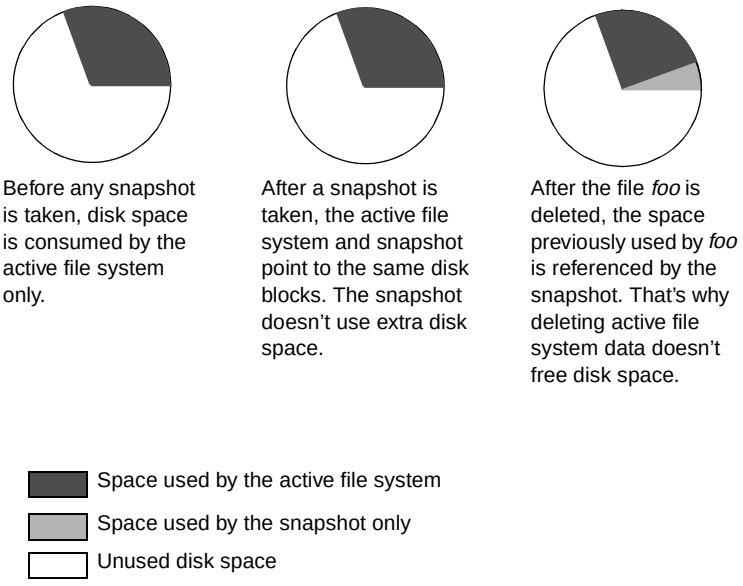


Figure 9-2. How snapshots use disk space

Changing the contents of *foo* creates a similar situation. New data written to *foo* cannot be stored in the same disk blocks as the current contents because the snapshot is using those blocks to store the old version of *foo*. Instead, the new data is written to new disk blocks, so there are two separate copies of *foo* on disk—a new copy in the active file system and an old one in the snapshot. This technique of duplicating disk blocks only as they are modified is called *copy-on-write*.

In summary, when the filer creates a snapshot, it doesn't use any disk space, but as files in the active file system are changed or deleted, the snapshot uses more and more disk space. How often files are changed and deleted determines the number of snapshots the filer can create and the length of time the snapshots can be kept.

In some directories, most data remains unchanged from day to day. For example, a user with a 10-MB home directory might change only 100 KB to 500 KB on a typical day. When files change slowly, snapshots can be kept on-line for days or even weeks before they begin to consume unacceptable amounts of disk space. In other directories, data changes quickly. If a large percentage of data changes every day, there might not be room to keep snapshots for even a few hours. To accommodate the needs of different users, create multiple volumes on the filer. In this way, you can apply different snapshot schedules to different volumes.

Understanding snapshot disk consumption

A confusing aspect of snapshots is the amount of disk space they currently consume and the amount of disk space they are likely to consume. The following sections explain with examples how to determine the amount of disk space used by snapshots.

Disk consumption by multiple identical snapshots

Suppose a snapshot contains a 1-MB file that hasn't changed since the filer created the snapshot. If that file is removed from the active file system, the snapshot then consumes 1 MB of disk space. The same version of that 1-MB file might be referenced by several different snapshots: *hourly.0*, *hourly.1*, and *hourly.2*. If these snapshots all contain the 1-MB file that hasn't changed since the filer created those snapshots, only 1 MB of disk space is consumed by the snapshots even though all three snapshots contain the file.

How the df command displays snapshots

To provide information about snapshot disk utilization, the `df` command on the filer treats snapshots as a partition different from the active file system. Following is a partial sample `df` command output:

```
toaster> df
Filesystem            kbytes    used    avail   capacity
/vol/vol0             3000000  2000000  1000000  65%
/vol/vol0/.snapshot   1000000 500000    500000   50%
```



NOTE: The numbers in this example were rounded off to make the example easier to understand. Also, to make the output easier to read, the "Mounted on" column is not included in the sample `df` output in the following sections.

In this example, the *vol0* volume contains 4 GB of disk space. It has 1 GB (or 25%) reserved for snapshots (the idea of reserving space for snapshots is described in more detail later). That leaves 3 GB for the active file system, and 2 GB of the file system is in

use. It is important to understand that the /vol/vol0/.snapshot line counts data that exists only in a snapshot. Because data that also exists in the active file system needs to be stored on disk anyway, it is misleading if the filer charged the space to snapshots. In the example, half of the 1 GB reserved for the snapshot is used.

How the snapshot reserve works

By default, the snapshot reserve is 20% of disk space. For information about how to adjust the amount of the snapshot reserve, refer to "Changing the snapshot reserve" on page 204.

This section uses several examples to explain the advantages of reserving disk space for snapshots.

If the filer created a snapshot when the disks were full, removing files from the active file system wouldn't create any free space, because everything in the active file system would also be referenced by the newly created snapshot. The filer would have to delete the snapshot before it could create any new files.

The following example shows how disk space being freed by deleting files in the active file system ends up in the snapshot:

If the filer creates a snapshot when the active file system is full and there is still space remaining in the snapshot reserve, the df command output is as follows:

```
Filesystem              kbytes    used   avail   capacity
/vol/vol0/              3000000 3000000       0     100%
/vol/vol0/.snapshot     1000000  500000  500000      50%
```

If you remove 100 MB of files, the disk space used by these files is no longer part of the active file system, so the space is charged to the snapshots instead. If you enter the `df` command, the output is as follows:

```
Filesystem               kbytes    used   avail   capacity
/vol/vol0/               3000000 2900000  100000       97%
/vol/vol0/.snapshot      1000000  600000  400000       60%
```

The filer reassigns 100 MB of space from the active file system to the snapshot reserve. Because there was reserve space for snapshots, removing files from the active file system freed space for new files.

Even with the snapshot reserve, the job of administering snapshot disk space consumption is important. There is no way to prevent snapshots from consuming disk space greater than the amount reserved for them. Consider what would happen in the example if all files in the active file system were deleted. Before the deletion, the `df` output was as follows:

```
Filesystem               kbytes    used   avail   capacity
/vol/vol0/               3000000 3000000       0      100%
/vol/vol0/.snapshot      1000000  500000  500000       50%
```

After removing all the data in the file system, the `df` command generates the following output:

```
Filesystem               kbytes    used   avail   capacity
/vol/vol0/               3000000 2500000  500000       83%
/vol/vol0/.snapshot      1000000 3500000       0      350%
```

The entire 3 GB in the active file system moved into snapshots, along with the 500 MB that were in snapshots before, making a total of 3.5 GB of snapshot data. This is 2.5 GB more than the

space reserved for snapshots. Because the active file system cannot use that space, the space shows up as used by the active file system even though no files are there.

Whenever snapshots consume more than 100% of the snapshot reserve, the system is in danger of becoming full. In this case, you can create files only after you remove enough snapshots. For example, if 500 MB of data were added to the active file system in the preceding example, a `df` command generates the following information:

```
Filesystem              kbytes    used   avail capacity
/vol/vol0               3000000 3000000       0     100%
/vol/vol0/.snapshot     1000000 3500000       0     350%
```

As soon as the filer creates a new snapshot, every block in the system is referenced by some snapshot. Therefore, no matter how many files you remove from the active file system, there is still not room to add any more. The only way to recover from this situation is to remove enough snapshots to free more disk space. Refer to "Displaying snapshot statistics" on page 201 for information about how to use the `snap list` command to determine which snapshot to delete.

Managing snapshot disk consumption

This section describes how to schedule snapshots to suit your environment, how to determine a reasonable snapshot reserve, and how to adjust the amount of space snapshots use. The examples in this section are for a volume named home.

Scheduling snapshots

The best way to manage the amount of space consumed by snapshots in each volume is to use `snap sched` to adjust the schedule of snapshot creation. Following are some suggested strategies for scheduling and retaining snapshots:

- If users rarely lose files or typically notice lost files right away, use the default snapshot schedule. For example, this is the schedule that creates a snapshot every day and keeps two:

  ```
  snap sched home 0 2 6@8,12,16,20
  ```

- If users commonly lose files and need to restore them, Network Appliance recommends that you delete the snapshots less often than you would in the preceding example. On many systems only 5% or 10% of the data changes each week, so the snapshot schedule of six nightly and two weekly snapshots consumes 10% to 20% of disk space. Considering the benefits of snapshots, it is worthwhile to reserve this amount of disk space for snapshots. Following is the recommended snapshot schedule, which keeps six daily snapshots and two weekly snapshots:

  ```
  snap sched home 2 6 8@8,12,16,20
  ```

- If the data changes very quickly, reduce the number of snapshots scheduled. For example, if a volume is filled and emptied each day (for example, a volume storing large temporary files for a CAD application), it might not make sense to use daily or weekly snapshots at all. On a very active volume, schedule snapshots every hour and keep them for just a few hours, or turn off snapshots. For example, the following schedule creates a snapshot every hour and keeps three:

  ```
  snap sched home 0 0 3
  ```

This schedule doesn't consume much disk space, and it lets users recover files in recent snapshots as long as they notice their mistake within a couple of hours.

Displaying snapshot statistics

The `snap list` command shows the amount of disk space used by snapshots in a specified volume. This command enables you to see how much disk space each snapshot uses, and helps you determine an appropriate snapshot reserve. Following is an example of the command output. If you don't specify a volume name in the command, the output contains statistics about each volume.

```
toaster> snap list vol0
Volume vol0
%/used       %/total    date          name
----------  ----------  ------------  --------
0% ( 0%)    0% ( 0%)    Jan 19 08:01  hourly.0
1% ( 1%)    1% ( 1%)    Jan 19 00:01  nightly.0
2% ( 2%)    2% ( 2%)    Jan 18 20:01  hourly.1
3% ( 2%)    2% ( 2%)    Jan 18 16:01  hourly.2
3% ( 2%)    3% ( 2%)    Jan 18 12:01  hourly.3
5% ( 3%)    4% ( 3%)    Jan 18 00:01  nightly.1
7% ( 4%)    6% ( 4%)    Jan 17 00:00  nightly.2
8% ( 4%)    7% ( 4%)    Jan 16 00:01  nightly.3
10%( 5%)    9% ( 4%)    Jan 15 00:01  nightly.4
```

- The "%/used" column shows space consumed by snapshots as a percentage of disk space being used in the volume. The first number is cumulative for all snapshots listed so far, and the second number is for the specified snapshot alone.

- The "%/total" column shows space consumed as a percentage of total disk space in the volume.

The "%/used" is more useful for planning the snapshot reserve because it is more likely to remain constant as the file system fills.

The following list explains in detail the percentages in the `snap list` output:

- The "%/used" column:

  The first value is equal to

  $$\frac{\text{cumulative snapshot space x 100\%}}{\text{cumulative snapshot space + file system space}}$$

  The second value is equal to

  $$\frac{\text{this snapshot x 100\%}}{\text{this snapshot + file system space}}$$

- The "%/total" column:

  The first value is equal to

  $$\frac{\text{cumulative snapshot space x 100\%}}{\text{total disk space in this volume}}$$

  The second value is equal to

  $$\frac{\text{this snapshot x 100\%}}{\text{total disk space in this volume}}$$

"Cumulative snapshot space" is the total space used by this snapshot and all other more recent snapshots (the ones preceding this snapshot in the `snap list` output).

This example shows a volume that keeps five nightly snapshots and four hourly ones. That is, the volume uses the following command for creating snapshots regularly:

```
snap sched vol0 0 5 4@8,12,16,20
```

The `snap list` output shows that the overhead for snapshots is only 10%, so the default snapshot reserve of 20% seems to be a waste of disk space. Assuming that this pattern of change holds up, a reserve of 12% to 15% provides a safe margin to ensure that removing files frees disk space when the active file system is full.

The values in parentheses (the values that show the space used by an individual snapshot) are useful in identifying a particular snapshot to remove when the file system is full. However, deleting a particular snapshot doesn't necessarily release the total amount of disk space indicated, because other snapshots might be referring to the same blocks. Refer to "Freeing disk space by deleting a snapshot" on page 204 for further information about how to select a snapshot file for deletion if you want to reclaim disk space.

If you do not want the total amount of disk space consumed by all snapshots to exceed a certain percentage of the used disk space, use the cumulative values in the `snap list` output to determine which snapshots to delete. In the preceding example, if you don't want more than 5% of used disk space to be spent by snapshots, delete all snapshots listed below *nightly.1* in the `snap list` output (that is, *nightly.2*, *nightly.3*, and *nightly.4*). After deleting the snapshots, *nightly.1* and all the other more recent snapshots consume 5% of the used disk space.

Changing the snapshot reserve

The snapshot reserve can be used only by snapshots, not by the active file system.

The default snapshot reserve is 20% of the available disk space. To change the reserve, enter a command in the following format:

`snap reserve` *`volume_name percent`*

For example:

```
snap reserve vol0 25
```

With no arguments, the `snap reserve` command displays the percentage of disk space reserved for snapshots in each volume.



NOTE: Snapshots can exceed the snapshot reserve space.

Freeing disk space by deleting a snapshot

This section describes how to use the `snap list` output to determine which snapshot file to delete to free the most disk space.

In the sample `snap list` output in "Displaying snapshot statistics" on page 201, the cumulative disk space used by snapshots gradually increases from top to bottom. For example, in the "%/used" column, the cumulative space used by *hourly.1* is 2% and the cumulative space used by *hourly.2* is 3%. This is not always the case. Consider a filer with a 100-MB file system that has not changed since the first snapshot was taken. The `snap list` command on this filer displays the following output:

```
%/used      %/total     date          name
--------    ---------   ------------  ---------
0% (0%)     0% (0%)     May 05 16:00  hourly.0
0% (0%)     0% (0%)     May 05 12:00  hourly.1
0% (0%)     0% (0%)     May 05 08:00  hourly.2
```

The cumulative disk space used by snapshots does not increase because no changes were made to the file system. However, if you had deleted 20 MB from the file system before the filer took the *hourly.0* snapshot, the `snap list` command would have displayed the following output:

```
%/used       %/total      date          name
--------     ---------    ------------  ---------
0%  (0%)      0% (0%)     May 05 16:00  hourly.0
20% (20%)     1% (1%)     May 05 12:00  hourly.1
20% (20%)     1% (1%)     May 05 08:00  hourly.2
```

In the "%/used" column, the cumulative values for *hourly.1* and *hourly.2* are both 20%, but the cumulative value for *hourly.2* is not 40%. This is because both snapshots point to the same 20 MB of data, the data that you just deleted.

The cumulative values for *hourly.1* and *hourly.2* are different if you delete and create data between snapshots in the following way:

1. Delete 20 MB of data and create 20 MB of new data after *hourly.2*.

2. Delete the 20 MB of data created in Step 1 after *hourly.1*.

After the data deletions and additions, the `snap list` command displays the following output:

```
%/used       %/total      date          name
--------     ---------    ------------  ---------
0%  (0%)      0% (0%)     May 05 16:00  hourly.0
20% (20%)     1% (1%)     May 05 12:00  hourly.1
33% (20%)     2% (1%)     May 05 08:00  hourly.2
```

In this scenario, *hourly.1* and *hourly.2* each consume 20% of the used disk space (20 MB out of 100 MB). However, this time they reference different data blocks. Cumulatively, they consume 40 MB, which is about 33% of the disk space used (120 MB, which is 40 MB used by snapshots plus 80 MB in the file system).

Before trying to conserve space by deleting a large snapshot file, examine the cumulative values in the `snap list` output. If two adjacent snapshot files show little difference in the cumulative values, most of the data referenced by the snapshots is the same. In this case, removing one of the snapshots doesn't free much disk space.

If you find snapshots confusing and hard to manage, use the default snapshot schedule and the default snapshot reserve because these settings are appropriate for most environments. When you create a new volume on a filer, remember that the new volume inherits the snapshot schedule from the root volume. After you use the volume for a while, check how much disk space the snapshots consume in the volume. If the disk space seems high, decrease the amount of time that snapshots are kept or increase the snapshot reserve. As you use snapshots, continue to watch the statistics change over time. The statistics help you gain a better understanding of how snapshots work.

Accessing snapshots from clients

Snapshots can be accessed by any user with the appropriate permissions. Every directory in the filer's active file system contains a directory named *.snapshot*, through which users can access old versions of files in that directory. How users gain access to snapshots depends on the file-sharing protocol used.

Obtaining access to snapshots from an NFS client

Figure 9-3 shows the directory structure on a client with the *vol0* volume of a filer named toaster mounted on */n/toaster*.

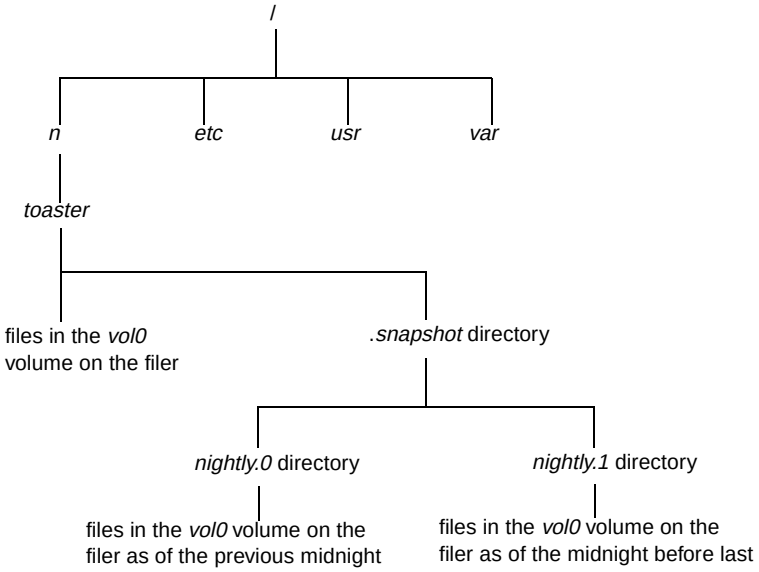


Figure 9-3. Directory tree on an NFS client with *vol0* mounted

In this example, the client can obtain access to snapshots by way of */n/toaster/.snapshot*. This might seem to contradict the explanation of snapshot access in the preceding section, because it shows a *.snapshot* directory only at the mount point instead of in every directory in the tree.

Actually, the *.snapshot* directory in the mount point is "real" to make the `pwd` command work, whereas the *.snapshot* directories in all other directories are "magic." A "magic" directory is one that can be accessed when it is referenced by name but does not show up in a directory listing. For example, at the mount point of a filer file system, a directory listing looks like this:

```
% ls -a
.      ..      .snapshot      dir1      dir2
```

The same command entered in a directory below the mount point does not show the *.snapshot* directory. For example:

```
% cd dir1
% ls -a
.      ..     file1      file2
```

If you enter the `ls` command with the directory name *.snapshot*, you can see a listing of the snapshots for the *dir1* directory:

```
% ls .snapshot
hourly.0      hourly.4      nightly.0      nightly.4
hourly.1      hourly.5      nightly.1      nightly.5
hourly.2      hourly.6      nightly.2      weekly.0
hourly.3      hourly.7      nightly.3      weekly.1
```

If *.snapshot* were to show up in every directory, it would cause many commands to work improperly. For instance, all recursive commands for removing files would fail because everything below *.snapshot* is read-only. Recursive copies would copy everything in the snapshots as well as files in the active file system, and a `find` command would generate a list much longer than expected.

Obtaining access to snapshots from a CIFS client

To CIFS clients, the snapshot directory appears only at the root of a share. For example, if a user's home directory is a share named bill, which corresponds to the */vol/vol0/home/bill* directory, only the */vol/vol0/home/bill/.snapshot* directory is visible. When this user displays the contents of the home directory, the snapshot

directory is displayed as *~snapshot* if the operating system supports long file names and as *~snapsht* if the operating system supports only short file names.



NOTE: The snapshot directory is visible in that it appears in a directory listing or File Manager display if the client operating system is configured to show hidden files.

In each directory within the share, a snapshot directory exists but is not visible to clients. For example, if the client operating system supports long file names, the applications on that operating system can use the snapshot at each level of the share by using *.snapshot*, *~snapshot*, or *~SNAPSHT* as the directory name. You cannot, however, display the directory name in any listing.

Determining snapshot versions

From an NFS client, the best way to find all versions of a particular file preserved in snapshots is to use the `ls` command. The following example shows how to find all versions of *foo*:

```
% ls -l foo .snapshot/*/foo
-rw-r--r--  1 smith 0 Jan 14 09:40  foo
-rw-r--r--  1 smith 0 Jan 13 18:39  .snapshot/nightly.0/foo
-rw-r--r--  1 smith 0 Jan 12 19:17  .snapshot/nightly.1/foo
```

The version of *foo* in the active file system was last modified on January 14, but the old versions available in the snapshots were modified on January 13 and January 12. Although users can use standard UNIX commands to examine the saved versions of *foo*, they cannot modify or remove these older versions because everything beneath *.snapshot* is read-only.

From a CIFS client, use the `Find` command to search for the file in the *~snapshot* directory. For example, if a user maps the home share to the F: drive and wants to find all versions of *foo* in snapshots, the user can use the `Find` command to search for *foo* in the *f:\~snapshot* folder.

Determining access times

When the filer creates a snapshot, the access time of each file in the snapshot is updated to the snapshot creation time. Thus, from an NFS client, you can use the `ls -lu` command, which shows the access times of files, to determine when snapshots were created. Following is an example of the `ls -lu` command:

```
% ls -lu foo .snapshot/*/foo
-rw-r--r--  1 smith 0 Jan 14 09:40  foo
-rw-r--r--  1 smith 0 Jan 14 00:00  .snapshot/nightly.0/foo
-rw-r--r--  1 smith 0 Jan 13 00:00  .snapshot/nightly.1/foo
```

From a CIFS client, you can determine the access time of a file by checking its properties.

Effects of snapshots on quotas

Quotas do not count disk space consumed by snapshots. If snapshots were included in the quota calculations, users could end up in a state where they could not create any new files until all snapshots containing their old files expired.

Snapshot options

The following options for the `vol options` command affect snapshots in the specified volume. The options remain in effect after the filer reboots.

`nosnap` Disables automatic snapshots. By default, this option is disabled.

`nosnapdir` Makes the *.snapshot* directory that is present at client mount points or the root of the CIFS share invisible. It also turns off access to the *.snapshot* directory and all *.snapshot* directories under the mount point or the root of the CIFS share. By default, this option is disabled.



NOTE: The `dump` command does not work if the `nosnapdir` or `nosnap` option is on.

Chapter 10: Managing Quotas and Maximum Number of Files

This chapter describes how to manage disk space on the filer by using disk quotas and by increasing the maximum number of files in a volume. This chapter also describes how to use the `df` command to monitor the current disk space usage.

Restricting disk usage by using disk quotas

Filer disk quotas restrict disk space and the number of files used by a given user or group. Quotas can also restrict disk space and the number of files used in a quota tree. A quota tree is a directory under the root directory of a volume. It is created by the `quota qtree` command.

You manage disk quotas by editing the */etc/quotas* file and using the `quota` command.

Format of the quotas file

To set up disk quotas, create a *quotas* file in */etc*. Following is a sample *quotas* file; the entries are explained later in this section.

```
#Quota Target       type                 disk     files
/vol/home/user/joe  user                 500M      10K
21                  group                750M      75K
/vol/eng/proj1      tree                 750M      75K
writers             group@/vol/eng/proj1 300M      50K
*                   user                 50M       10K
```



NOTE: Keep a record of your *quotas* file in a safe place and update it as you change it in case you must do a restore without having access to the root volume.

Quota Target field

The Quota Target field specifies the user, group, or tree on which you want to impose restrictions. You can assign more than one quota to a user or group, but only one quota to a tree. The entries can be in any order.

Quota target for a user quota. You specify a user with one of the following targets:

- a file or subdirectory whose UID matches the user
- the user's user name, as defined in the */etc/passwd* file or the NIS password map
- the user's UID

The methods are equivalent, and inform the filer of the UID of the target. A file or directory is used only as the source of a UID; there are no quota implications for that file or directory. The UID of the user must not be 0.

Any file or subdirectory you use in the Quota Target field is referenced repeatedly throughout the life of the system, so if you use a path name, choose a path name that will last for as long as the user account remains on the system. For example, use a user's home directory for a user quota.

Quota target for a group quota. You specify a group with one of the following targets:

- a file or subdirectory whose GID matches the group
- the group's name
- the group's GID

The methods are equivalent. A file or directory is used only as the source of a GID; there are no quota implications for that file or directory. The GID of the group must not be 0.

Quota target for a tree quota. To create a tree quota, use the `quota qtree` command to create a directory in the root directory of a volume. The quota target in the *quotas* file for a tree quota is the complete path name of this directory.

Quota target for default quotas. Use an asterisk (*) in the Quota Target field to specify a default for the user or group quotas. (Defaults do not apply to tree quotas.) The default value applies to the following users or groups:

- New users or groups that are created after the default entry has taken effect. For example, if the maximum disk space for default user quotas is 500 MB, any new user can use up to 500 MB of disk space.
- Users or groups that are not explicitly mentioned in the *quotas* file. For example, if the maximum disk space for default user quotas is 500 MB, users for whom you have not specified a user quota in the *quotas* file can use up to 500 MB of disk space.

To override a default for a specific user or group, specify a quota for that user or group.

Type field

You can enter one of the following values in the Type field to define the quota type:

- user. If a user quota applies just to a tree and not to the entire volume, specify user@*tree* where *tree* is the name of a quota tree. If a user quota applies to a volume other than the root volume, append `@/vol/`*`volume`* to the quota type. For example, `user@/vol/marketing` means that the user quota applies to the *marketing* volume.

- group. If a group quota applies just to a tree and not the entire volume, specify group@*tree* where *tree* is the name of a quota tree. If a group quota applies to a volume other than the root volume, append *@volume* to the quota type. For example, `group@/vol/marketing` means that the group quota applies to the *marketing* volume.

- tree. A tree quota is similar to a disk partition, but you can increase or decrease the size of a tree quota at any time.

Disk field. It specifies the maximum amount of disk space that the quota target can use. In this field, K is equivalent to 1024 bytes, M means 2^20 bytes, and G means 2^30 bytes. If you omit the K, M, or G, the default is K.



NOTE: Do not put a blank in the disk field. It acts as white space. The filer regards the following entries as equivalent:

```
#Quota Target      type      disk      files
/export            tree                75K
/export            tree      75K
```

Files field. It specifies the maximum number of files that the quota target can use. Use K to indicate 1024 files. For example, 75K means 76,800 files. Use M to mean 2^20 and G to mean 2^30. You can omit the K, M, or G. For example, if you enter 100, it means

the maximum number of files is 100. A blank in this field means there is no restriction on the number of files that the quota target can use.

Sample quotas file

Following is a sample *quotas* file that includes different kinds of quotas:

```
#Quota Target          type                  disk      files
/vol/home/user/jdoe    user                  500M      10K
108                    user                  500M      10K
jsmith                 user@/vol/rls         500M      10K
publications           group                 750M
/vol/home/eng          group@/vol/cad        750M      75K
/vol/cad/proj1         tree                  750M      75K
writers                group@/vol/cad/proj1  150M
*                      user                   50M      15K
*                      user@/vol/cad/proj1    50M      10K
*                      group                 750M      85K
*                      group@/vol/cad/proj1  100M      75K
mhoward                user                  150M     100K
mhoward                user@/vol/cad/proj1    75M      75K
```

Any operation that creates files or writes to them must satisfy all applicable quotas. The following list describes the effects of the sample *quotas* file:

- The owner of */vol/home/user/jdoe* and the user whose UID is 108 can each use 500 MB of disk space and 10,240 files in the root volume.
- The user whose user name is jsmith can use 500 MB of disk space and 10,240 files in the *rls* volume.
- The group publications can use 750 MB of disk space with no restrictions on the number of files in the root volume.
- The group that owns */vol/home/eng* can use 750 MB of disk space and 76,800 files in the *cad* volume.

- The quota tree proj1 in the *cad* volume can use 750 MB of disk space and 76,800 files.
- The writers group can use 150 MB of disk space and an unlimited number of files in the proj1 tree provided that the quotas on the proj1 tree are not exceeded.
- Any user not otherwise mentioned in this file can use 50 MB of disk space and 15,360 files in the root volume.
- In the proj1 tree, any user not otherwise mentioned in this file can use 50 MB of disk space and 10,240 files.
- Any group not otherwise mentioned in this file can use 750 MB of disk space and 87,040 files in the root volume.
- In the proj1 tree, any group not otherwise mentioned in this file can use 100 MB of disk space and 76,800 files.
- The user mhoward can use 150 MB of disk space and 102,400 files in the root volume. In the proj1 tree, mhoward can use 75 MB of disk space and 76,800 files.

Both user and group quotas apply to the entire specified volume (or the root volume if no volume name is specified). This is true even if the quota target in the *quotas* file is specified in the form of a path name. For example, if the quota target for the user named jdoe is */vol/home/user/jdoe*, the filer imposes the quotas on all files written by jdoe, not just the ones written to */vol/home/user/jdoe*.

The quota command

The *quotas* file specifies what the restrictions are on users, groups, and trees. Whether these restrictions take effect depends on the `quota` command. The `quota` command enables you to do the following tasks:

- enable and disable quotas on a per-volume basis
- resize quotas on a per-volume basis

- display information about all active quotas or about quotas that apply to a specified path
- create a quota tree

Some of the information from `quota` commands is available through SNMP using the Network Appliance custom MIB. For more information about the MIB, refer to "New Network Appliance custom MIB" on page 82.

Enabling or disabling quotas

Use the following command syntax to enable quotas for a volume:

```
quota on [ volume ]
```

This command computes the disk usage of each quota target. The computation can take a few minutes to complete for a large number of quotas. To find out how much quota initialization the filer has completed, enter the `quota` command without any arguments. For example:

```
toaster> quota
vol0: quotas are on.
vol1: quotas are initializing (24% done).
vol3: quotas are off.
```

Use the following command syntax to disable quotas for a volume:

```
quota off [ volume ]
```

Because the filer remembers whether quotas are on or off even after it reboots, there is no need to add a `quota` command to */etc/rc*.

Resizing quotas

The `quota resize` command updates active quotas without recalculating disk usage, and is faster than `quota off` followed by `quota on`. An active quota is one that appears in the output of the

quota report command discussed in "Displaying information about quotas" on page 221. You use the quota resize command after changing limits for a group or user. For example, jdoe has a disk quota of 500 MB, as shown in the following example:

```
#Quota Target          type          disk      files
/vol/home/user/jdoe    user          500M      10K
*                      user           10M      10K
```

Use the quota resize command to update the quota if jdoe's disk quota were increased, as shown in the following example:

```
#Quota Target          type          disk      files
/vol/home/user/jdoe    user          600M      10K
*                      user           10M      10K
```

How quota resize affects newly added quota targets

The quota resize command usually ignores newly added quota targets. For each entry that it skips, the quota resize command prints the following message:

`quota resize: new entry on line` *n* `in /etc/quotas`

where *n* is a line number.

The quota resize command does not ignore newly added quota targets in the following situation:

If a default quota applies to the creator of a file, an active quota record is created for the owner of the file. If a newly added quota target is the user or group that has an active quota record, the quota resize command does not ignore the newly added quota target. That is, if a user or group has written to a file that is under the control of a default quota, a newly created entry in the *quotas* file for this user or group takes effect after a quota resize command.

After you edit the *quotas* file, if you want to make sure that all entries take effect, enter the `quota off` command followed by the `quota on` command. In this way, all quotas listed in the file become active quotas.

Creating an active quota for a quota target

This section describes how to use the `quota resize` command so that quotas can take effect on targets that have not created any files.

Because a quota becomes active when a quota target has written a file, you can make an entry in the *quotas* file an active quota by following these steps:

1. Create a file.
2. Change ownership of the file to the quota target.

Because the quota target now has an active quota record, you can enter the `quota resize` command, including the volume name, to make the quota entry go into effect. This procedure takes less time than executing the `quota off` command followed by the `quota on` command because only the newly active quota is recalculated.

Displaying information about quotas

The `quota report` command displays the current consumption of files and space for each quota target and for each user and group that is under quota restrictions.

For example, if a user quota is specified in the *quotas* file for a user named jdoe, the quota report shows how many files and how much disk space have been used by jdoe. The Quota Specifier column in the quota report shows the same information as the Quota Target column in the *quotas* file, with the following exception:

If a user or group is under default quota restrictions, `quota report` displays information about the user or group as if the user or group had an entry in the *quotas* file. In this case, the Quota Specifier column in the quota report is blank.

With a path argument, `quota report` displays information about all quotas that apply to the specified file.

Creating a quota tree

The `quota qtree` command, with no arguments, lists the quota trees on the filer. The list differs from the output from `quota report` in that the list includes all quota trees created, regardless of whether they were entered into the *quotas* file. Any directory that was created by `quota qtree` is a quota tree, regardless of whether you specified quota restrictions on the tree in the *quotas* file.

Creating a user quota

Follow these steps to limit disk space used by a user named jdoe:

1. Decide who should be limited by a disk quota. In this example, the user is the owner of */vol/home/user/jdoe*. The filer restricts disk usage of files with the same UID as */vol/home/user/jdoe*.

2. Add the following line to the *quotas* file:

   ```
   /vol/home/user/jdoe  user  300M  20K
   ```

   Substitute the values you want for 300M and 20K. You can also use the UID of jdoe or the name jdoe if you have set up a *passwd* file on the filer or the NIS database. If you want the

restrictions to apply to a volume other than the root volume (for example, the *home* volume), enter the following line:

```
/vol/home/user/jdoe  user@/vol/home  300M  20K
```

3. Use the quota on command, including the volume name. For example, if quotas are already on for the *home* volume, enter the following commands:

```
quota off home
quota on home
```

Alternatively, if jdoe is already under quota restriction, for example, if his files were restricted by a default user quota, enter the following command so that the user quota you just created can take effect:

```
quota resize home
```

Creating a group quota

Follow these steps to limit disk space used by a group named service:

1. Decide which group should be limited by a disk quota. In this example, the group is the owner of */vol/home/user/service*. That is, the filer restricts disk usage of files with the same GID as */vol/home/group/service*.

2. Add the following line to the *quotas* file:

```
/vol/home/group/service  group  700M  100K
```

Substitute the values you want for 700M and 100K. You can also use the GID of the group or the name of the group. If you want the restrictions to apply to a volume other than the root volume, for example, the *home* volume, enter the following line:

```
/vol/home/group/service  group@/vol/home  700M  100K
```

3. Use the `quota on` command. If quotas are already on for the *home* volume, enter the following commands:

```
quota off home
quota on home
```

Alternatively, if the service group is already under quota restriction, for example, if the group's files are already restricted by a default group quota, enter the following command so that the group quota you just created can take effect:

```
quota resize home
```

Creating a quota tree

A quota tree limits the amount of space consumed by a directory, regardless of the user or group ownership of the directory's contents. You cannot impose tree quotas on directories that were created by a command on a client (for example, the `mkdir` command on an NFS client). You can impose tree quotas only on directories that were created by the `quota qtree` command, as in the following procedure:

1. Use the following command syntax to create a quota tree:

   `quota qtree` *`pathname`*

   You can create a quota tree only in the root directory of a volume. If *pathname* does not begin with a slash (/), the quota tree is created in the root volume. For example:

   ```
   quota qtree news
   ```

   creates the quota tree */vol/vol0/news*.

   If you want to create a quota tree in a particular volume (for example, *home*), enter the following command:

   ```
   quota qtree /vol/home/news
   ```

NOTE: By default, every user is given read, write, and execute permissions to the quota tree. Remember to change the permissions if you want to restrict access to the tree for some users.

2. Add the following line to the *quotas* file if the tree is in the root volume:

   ```
   /news  tree  400M  50K
   ```

   Add the following line to the *quotas* file if the tree is in the *home* volume:

   ```
   /vol/home/news  tree  400M  50K
   ```

3. If you want to restrict disk usage by users or groups in the quota tree, add the user or group quotas. For example, to limit jdoe to 10 MB of disk space and 10,240 files in the */vol/home/news* quota tree, add the following line in the *quotas* file:

   ```
   jdoe    user@/vol/home/news   10M   10K
   ```

4. Use the `quota on` command. If quotas are on already, enter the following commands:

   ```
   quota off home
   quota on home
   ```

CAUTION: You cannot move a file into or out of a quota tree with a rename operation. As a result, the `mv` command on some UNIX systems fails if you try to move a file into or out of a quota tree. You can always move the file by copying it, then deleting the original.

Removing quota restrictions

To remove quota restrictions, follow these steps:

1. Remove the appropriate line or lines from the *quotas* file.

2. Use the `quota off` and `quota on` commands, including the volume name. For example:

```
quota off home
quota on home
```

When quotas are exceeded

This section describes how the filer responds when quotas are exceeded and what users see on the client systems.

Messages displayed by the filer when quotas are exceeded

When it receives a write request, the filer first determines whether the file to be written is in a quota tree. If the write would exceed the tree quota, the filer logs the following error message:

```
tid tree_id: tree quota exceeded on volume_name
```

If the quota tree is not full but the write would cause either the user or group quota to be exceeded, the filer logs one of the following errors:

```
uid user_id: disk quota exceeded
gid group_id: disk quota exceeded
```

where *user_id* is the user's UID and *group_id* is the group's GID.

To the client, the filer returns an "out of disk space" error to the NFS write request or a "disk full" error to the CIFS write request. The following sections describe how the clients notify the users about quotas being exceeded.

Messages displayed on NFS clients

If a write from an NFS client to a filer causes a quota to be exceeded, the user experience depends on the operating system version and the application.

If a UNIX client mounts a filer without the `noquota` option, the `login` program on the client checks to see whether the user has reached the disk quota and file quota each time the user tries to

log in to the client. The client displays a message to alert the user before displaying the system prompt if a quota was reached. In the following example, a user reached the disk quota on the filer mounted as */t/toaster* on a client named client2:

```
client1% rlogin client2
You have mail.
Block limit reached on /t/toaster
client2%
```

Not all versions of UNIX perform the quota check as described in this section. Also, the exact message printed varies from one version to another.

If a write causes a quota to be exceeded, the error message seen by the user depends on the application. For example, on a SunOS 4.x client, if a user tries to save a file using `vi` when his or her disk quota is reached, the error message is

```
Disc quota exceeded [Warning -
/t/toaster/home/jdoe/file1 is incomplete]
```

Messages displayed on CIFS clients

If a write from a CIFS client to a filer causes a quota to be exceeded, the user experience depends on the operating system and the application. Following are two examples:

- An application might display a message as follows:

  `Cannot write file` *`filename`*

- When a user tries to copy a file to the filer using the Explorer in Windows 95, the error is as follows:

  `Cannot create or replace` *`filename`*`: Cannot read from the source file or disk.`

Increasing the maximum number of files

Initially, the maximum number of files on the filer is set at one for every 32 KB of disk space. The number is increased automatically when you add a new disk. The increase is determined by the filer, and is not a user-specified value.

Unlike UNIX, which requires that you specify the maximum number of files in a file system when you create the file system, the filer enables you to use the `maxfiles` command to increase the number of files for each volume at any time.



NOTE: Use caution when increasing the maximum number of files because after you increase this number, you can never reduce it. As new files are created, the file system consumes the additional disk space required to hold the inodes for the additional files; there is no way for the filer to release that disk space. An inode is a data structure containing information about files.

To see how many files are in a volume, use the `df -i` *`pathname`* command, which shows how many inodes have been used, or use the `maxfiles` *`volume`* command. For example, both of the following commands show that the *home* volume has used 2,872 inodes:

```
toaster> df -i /vol/home
Filesystem     iused     ifree    %iused   Mounted on
/vol/home/      2872      118090      2%    /vol/home
toaster> maxfiles home
Volume home: maximum number of files is currently
120962 (2872 used)
```

Information generated by the `maxfiles` command is available through SNMP using the Network Appliance custom MIB. For more information about the MIB, see "New Network Appliance custom MIB" on page 82.

The df command

To verify the amount of free disk space on the filer, enter the `df` command on the filer. Information generated from the `df` command is also available through SNMP, using the Network Appliance custom MIB, which is described in "New Network Appliance custom MIB" on page 82.

With the `-i` option, the command displays the number of used inodes and the number of available inodes. Following is an example of the `df -i` command:

```
toaster> df -i /vol/home
Filesystem    iused      ifree    %iused   Mounted on
/vol/home/    240843     121525     66%     /vol/home
```

The total amount of disk space shown in the `df` output is less than the sum of available space on all disks installed in the volume. Consider the following sample `df` command issued on a volume with seven 2-GB disks installed:

```
toaster> df /vol/vol0
Filesystem          kbytes    used       avail   capacity
/vol/vol0/          9886448   8426196    1460252 85%
/vol/vol0/.snapshot 1098492   427008     671484  39%
```

When you add the numbers in the kbytes column, notice that the sum, 10,984,940 KB (about 11 GB), is significantly less than the total disk space installed (14 GB). The following reasons account for the discrepancy:

- The parity disk, which is a 2-GB disk in this example, is not reflected in the output of the `df` command.
- As with the UNIX FFS (Fast File System), the filer reserves 10 percent of the total disk space for efficiency, which `df` does not count as part of the file system space.

The filer doesn't limit the size of a file system. The maximum size of the file system is restricted only by the amount of disk space installed in the volume. For individual files, the maximum size is 4 GB minus one byte. (Many NFS clients restrict the file size to 2 GB.)

Using the df command with quota trees

When you enter a `df` command with a path name on a client, the command returns the amount of free space in the file system containing the path name. For example, if the filer is mounted on the client as */t/toaster*, the `df` command on the client displays the disk information about the */t/toaster* file system as follows:

```
client% df /t/toaster/engineering/jdoe
Filesystem           kbytes    used   avail capacity  Mounted on
toaster:/           2097151 1646923  450228    79%    /t/toaster
```

However, if you defined quota trees on the filer, the information about available space might be misleading because the actual space available might be less. For example, if */engineering* is a quota tree with a disk quota of 1,800 MB, the space available in the */t/toaster/engineering* directory is less than that in the `df` command output shown in the preceding example.

If you have quota trees on the filer, Network Appliance recommends that you mount each quota tree separately. For example, if the filer named toaster has two quota trees, */vol/home/engineering* and */vol/home/marketing*, mount *toaster:/vol/home/engineering* and *toaster:/vol/home/marketing* on two mount points (for example, */t/toaster/engineering* and */t/toaster/marketing*). In this way, the filer takes the quota trees into consideration when responding to a `df` command from a client and returns the amount of free space in each quota tree, as opposed to the space available in the entire file system.

Chapter 11: **Dump and Restore**

This chapter discusses the following topics:

- using the filer's `dump` and `restore` commands to back up files to tape (creating a system backup) and restore files from tape
- using a filer's tape drive from another system
- using NDMP (Network Data Management Protocol) support for third-party backup applications

The term *system backup* means backing up files from the filer to tape, not the creation of on-line, read-only copies of the entire file system (snapshots), which is described in Chapter 9, "Snapshots."

The term *active file system* means any file system on the filer, excluding snapshots.

If your filer is running *only* NFS, you can do a system backup using UNIX commands. If your filer is running *only* CIFS, you can do a system backup using CIFS utilities.



CAUTION: If your filer delivers both NFS and CIFS service, use only the filer `dump` and `restore` commands. If you use any other backup utilities, file system information such as 8.3 names, permissions, or DOS attributes might be missing or incorrect for the restored files. Furthermore, symbolic links and other types of files not known to CIFS clients might not be restored if backup and restore utilities are used on a CIFS client.

Backing up the filer to tape

If you are using NFS only, you can back up the filer using commands such as `tar` and `cpio` from a client system. However, use `tar` and `cpio` with caution because some versions have bugs that cause them to fail with file systems that contain long path names, unusual file names, or hard links.

A more efficient and reliable way of backing up the file system is to use the filer `dump` command, which creates backup files compatible with the SunOS 5.*x* / Solaris 2.*x* `ufsdump` format. You can use `dump` on a directory in an active file system (in which case `dump` creates a snapshot and dumps that) or on a snapshot. Before issuing the `dump` command, make sure that the `nosnap` and `nosnapdir` options are off for each volume to be backed up.



NOTE: If a file has a UID or GID longer than 16 bits, the filer uses the Solaris 2.5.1 scheme for putting that file on the dump tape. Files and directories with UIDs or GIDs longer than 16 bits restored on systems that use Solaris earlier than 2.5.1 might not be restored with the correct UIDs or GIDs.

Features of the filer's dump command

The `dump` command offers the following features:

- It can write to standard output, to a remote tape device on a host supporting `rmt` (remote magnetic tape protocol), or to a local tape device attached to the filer.
- It can back up an active subtree of volume, which can be a file, directory, or the entire volume.
- It supports backing up a large directory to multiple files on a single tape or multiple files on multiple tapes. With this feature, you can back up a directory to a remote host even if the directory size exceeds the maximum size allowed for a single tape file on the remote host. Each file on the tape is called a *tape volume*.



NOTE: A tape volume is a file on the backup tape, which is a part of an archive for a file system or subtree. It should be distinguished from a disk volume, which is a file system on the filer.

Planning for system backups

Before initiating the `dump` command, consider the following information and write down the requirements and limitations of your environment:

- If you dump to a remote host, check the maximum size of the dump file supported by the host. Some UNIX systems (for example, systems running SunOS 4.1.1 or earlier) don't support dump files larger than 2 GB. In the `dump` command, you can specify the dump file size in blocks to avoid exceeding the limit imposed by the remote host.

- If you dump to a tape attached to the filer, the dump file can be greater than 2 GB. If you use a host other than the filer for restoring the files, you must determine whether the host supports dump files larger than 2 GB. If it does not, limit the size of the dump file by specifying the appropriate dump file size in the `dump` command.

- Verify the blocking factor supported by the system on which the dump tape will be restored; some systems, including the filer, don't support blocking factors larger than 63 KB (the default). If you dump to a local tape drive, the filer restricts the blocking factor to 63 KB. However, if you dump to a remote host that allows a higher blocking factor, do not exceed the maximum blocking factor supported by the system to be used for restore.

- If you plan on restoring quota trees, make sure that every quota tree is listed in the */etc/quotas* file. If you want to restore your quota trees with your file system, you must re-create the quota tree before proceeding with the restore. The */etc/quotas* file provides an accurate list of the quota trees to re-create. For

information about quota trees, see “Creating a quota tree” on page 224.

CAUTION: Plan on verifying that dumps have been completed successfully. For example, if you are executing `dump` through the `rsh` command and `dump` needs more tapes than are available in your tape device, it displays the following messages (among others):

```
DUMP: ran out of tape, and can't ask questions when
run via "rsh".
DUMP: The ENTIRE dump is aborted.
```

If you did not check whether the dump succeeded and kept putting in new tapes and executing additional dumps, you would have tapes containing no dumps, and you might lose data.

Dump and restore for large volumes

If you did not set up multiple volumes on your filer such that each volume can be dumped and restored quickly, segment individual dumps so that they can be completed sooner.

To restrict dump trees to a predetermined size, use quota trees within your file system to make sure that directories in that file system do not grow beyond a predetermined size. For information about quota trees, see “Creating a quota tree” on page 224.

Determining the number of tapes required

To determine how many tapes you need for backing up a subtree, follow these steps:

1. Verify the size of the dump tape to determine the number of tapes required. Consult your tape drive documentation for information about compression. If you dump to a remote host, check the maximum capacity of the tape drive on that host. When calculating the number of tapes needed, remember that `dump` can back up to multiple tape volumes on multiple tape drives.

To determine the capacity of the tape drive attached to the filer, enter the `sysconfig -t` command. Following is an example for a filer containing a Quantum DLT7000 tape drive. The drive is attached to SCSI adapter 2 and has SCSI ID 4.

```
toaster> sysconfig -t
Tape drive (2.4)  Quantum DLT7000
rst0l - rewind device,         format is: 81633 bpi 20 GB
nrst0l - no rewind device,     format is: 81633 bpi 20 GB
urst0l - unload/reload device, format is: 81633 bpi 20 GB
rst0m - rewind device,         format is: 81633 bpi 40 GB
(w/comp)
nrst0m - no rewind device,     format is: 81633 bpi 40 GB
(w/comp)
urst0m - unload/reload device, format is: 81633 bpi 40 GB
(w/comp)
rst0h - rewind device,         format is: 85937 bpi 35 GB
nrst0h - no rewind device,     format is: 85937 bpi 35 GB
urst0h - unload/reload device, format is: 85937 bpi 35 GB
rst0a - rewind device,         format is: 85937 bpi 70 GB
(w/comp)
nrst0a - no rewind device,     format is: 85937 bpi 70 GB
(w/comp)
urst0a - unload/reload device, format is: 85937 bpi 70 GB
(w/comp)
```



NOTE: Stackers and libraries that are connected to the filer through SCSI, such as the Breece-Hill library, are not visible to the `sysconfig` command. Only the tape drive, as in the preceding example for the DLT7000, is visible.

2. To determine the sizes of various files in a file system to be backed up, follow these steps:

   a. Use the `quota report` command to list the usage in each of the quota trees to be dumped. For example:

```
toaster> quota report
                              K-Bytes          Files
Type  ID  Volume Tree     Used Limit      Used Limit      Quota Specifier
----- --- ----- ----      ---- -----      ----- -----     ---------------
tree  6   vol1   finance  21608 1048576   1350 10240     /vol/vol1/finance
user 1086 vol2   -          9172 1048576  129 1048576     /vol/vol2/eirik
tree  3   vol2   -       3018520 10485760 403451 1048576  /vol/vol1/mkt
tree  4   vol1   news    6247100 20971520 1561775 2097152 /vol/vol1/news
tree  1   vol1   users   3018328 10485760  406421 1048576 /vol/vol1/users
```

   b. If any directories to be backed up are not in the quota report (because they are not quota trees or they are unlisted quota trees), find out the size of each directory by checking its properties from a CIFS client, or use the `du -s` command from an NFS client.

      Following is an example of checking the disk space used by directories from an NFS client. In this example, the filer root directory (*/vol/vol0*) is mounted on */n/toaster* on a client.

```
client% cd /n/toaster

client% du -s etc home

4108 etc

5510100 home
```

      If each tape stores up to 10 GB of data and the host on which you dump and restore supports dump files larger than 2 GB, you can dump the *mkt* and *news* subtrees to one tape volume. Then load another tape and dump the rest of the file system (that is, the *etc*, *finance*, *home*, and *users* subtrees).

When you dump multiple subtrees on one tape, you must use the `norewind` tape device in the `dump` command. For information about the `dump` command, see "The dump command syntax" later in this chapter.

Supported tape drives

If you are using the tape drive on a client system, you can dump files to or restore files from any tape drive on the client.

If you need to add an external tape drive to your filer, you can purchase a drive from Network Appliance or another vendor. Refer to the hardware guide for your filer model for instructions about attaching an external tape drive and a list of supported tape drives; using an unsupported drive might cause the filer to hang.

The dump command syntax

Because the filer supports `rsh`, you can automate system backups through shell scripts. Backing up the filer using `rsh` offers another advantage over starting `dump` on the console: you can be sure that `dump` is not aborted by a Ctrl-C inadvertently entered from a host connected to the filer through `telnet`.

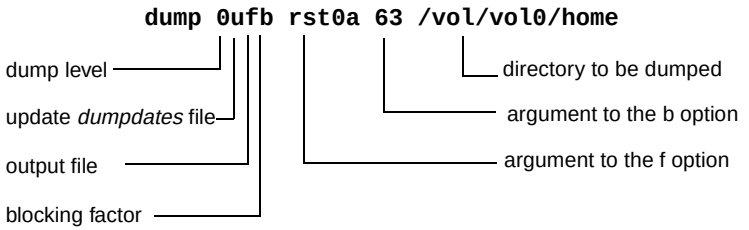
The information provided in this section applies whether the command is entered at the filer prompt or through `rsh`.

Following is the command syntax for `dump`:

`dump [ options [ arguments ]] [ subtree ]`

Specify multiple options with no intervening spaces, then the arguments to each option in the order of the options. You can append a subtree. If the subtree name does not include the volume name, `dump` assumes that the subtree is within the filer root volume. The options are listed in Table 11-1 on page 240.

A typical dump command looks like the following example:



The preceding example dumps one directory (*/vol/vol0/home*) on a local tape device (*rst0a*).

The filer dump command is similar to the Solaris ufsdump command. A major difference between the commands is that the blocking factor in the filer dump command is in kilobytes, while in the SunOS dump command, the blocking factor is in 512-byte units.

The *subtree* parameter specifies a file or directory that resides within the active file system or within a snapshot. If the subtree is within the active file system, the dump command creates a snapshot named *snapshot_for_dump.n* (where *n* is a number that is reset to 0 at boot time and then incremented by one after each *snapshot_for_dump.n*), from which it copies the data to tape. The filer automatically deletes *snapshot_for_dump.n* that it created when it is finished with the dump. If you are doing a dump of an existing snapshot, no *snapshot_for_dump.n* is created by that dump.

NOTE: If multiple dump commands run simultaneously, each dump creates its own *snapshot_for_dump.n*. For example, the following files might exist on your filer when several dump commands are taking place: *snapshot_for_dump.0*, *snapshot_for_dump.1*, and *snapshot_for_dump.2*. Run no more than four concurrent dumps.

When dumping multiple subtrees, you must also specify a dump name. For more information about dumping multiple subtrees, see “Dumping multiple subtrees” on page 243.

CAUTION: Do not try to delete a snapshot while `dump` is backing up the snapshot; this causes the filer to crash. If `dump` is invoked on a snapshot whose name allows the snapshot scheduler to delete it (an hourly, nightly, or weekly snapshot), the snapshot scheduler can also cause a crash by attempting to delete such a snapshot. To avoid such crashes, do one of the following:

- Ensure that the dump finishes before the snapshot being dumped comes up for deletion by the snapshot scheduler.
- Rename the snapshot to protect it from the snapshot scheduler.

Table 11-1 describes the options for the `dump` command.

Table 11-1. Options for the dump command

Option	Argument	Meaning
0–9		Specifies the dump level. Level 0 is a full backup. Use this option in the same way as you would on a UNIX system (for example, SunOS). If specified, this must be the first key. The default is 0.
u		Updates the */etc/dumpdates* file after a successful dump. This ASCII file contains information about each dump. Its format is identical to one created by the Solaris `ufsdump` command.

Table 11-1. Options for the dump command (continued)

Option	Argument	Meaning
f	*files*	Specifies the files to which `dump` writes. You can specify a comma-separated list of files, and each file is considered a dump volume. Following are legitimate file names: • local tape drive names (for example, `nrst0a`). For more information about tape drive names, refer to the *na_tape*(4) man page. • remote tape drive names, which must be specified with the host name (for example, `tapemachine:/dev/rst0,/dev/rst1`) • standard output, specified as a dash (–).
b	*factor*	Specifies the blocking factor in kilobytes. If you dump to local tape drives, use a blocking factor no greater than 63 KB. Make sure that the blocking factor you specify is supported on the system on which the dump file will be restored.
B	*blocks*	Specifies the maximum size of each dump volume in 1,024-byte (1-KB) blocks. When the subtree being dumped exceeds one volume, `dump` automatically writes to the next tape device specified with the `f` option. If there are no more files in the list, `dump` prompts you for a new tape to be loaded to the last device.[a] For example, to limit the dump volume size to 2 GB (so that the dump file can be restored on a UNIX system with a 2-GB limitation), specify 2,097,151 blocks. For information about uncompressed and compressed capacity of tape drives, consult your tape drive documentation. Use this option only if you plan to restore the dump volume on a host that has a size limit for a dump volume.

Table 11-1. Options for the dump command (continued)

Option	Argument	Meaning
l		Specifies a dump containing named subtrees of the directory specified in the final argument of the command line. The dump command reads in subtree names from standard input (entered one per line with the list terminated by one blank line) and dumps only those subtrees.
n	*name*	Specifies the name of the dump volume for */etc/dumpdates*. This is mandatory when dumping multiple subtrees.

a. Use the sysconfig -t command to display information about the capacity of a filer tape drive.

Following are several examples of the dump command for backing up the */vol/vol0/home* directory:

- To dump to one tape volume on a local tape device

  ```
  dump 0uf rst0a /vol/vol0/home
  ```

- To dump to one tape volume on a remote tape device

  ```
  dump 0uf adminhost:/dev/rst0 /vol/vol0/home
  ```

- To dump a directory to multiple tape volumes on a tape stacker or library without operator intervention

  ```
  dump 0uf urst0a,urst0a,urst0a,urst0a,rst0a
  /vol/vol0/home
  ```

- To back up a directory in an existing snapshot

  ```
  dump 0unf /home rst0a /vol/vol0/.snapshot/weekly.0/home
  ```

- To back up the entire root volume in sequential mode to a Breece-Hill library if four tapes are required

  ```
  dump 0uf urst0a,urst0a,usrst0a,nrst0a /vol/vol0/
  ```



NOTE: When you back up a directory in an existing snapshot, use the `n` option to ensure that the *dumpdates* entry is correct.

For information about using the tape drive on another filer to make backup tapes, see "Using a filer tape drive from another system" on page 257. For a list of the device names to use for the local tape drive, use the `sysconfig -t` command.

See the *na_dump*(1) and *na_tape*(4) man pages for more information about the `dump` command.

Dumping multiple subtrees

You can dump multiple subtrees with a single `dump` command by using

- the `n` option to specify the name of the dump volume for the */etc/dumpdates* file.
- the `l` option to specify that `dump` reads path names from standard input.
- a directory name as the last command argument. This directory contains the subtrees to be dumped.

After you enter the `dump` command, the filer displays some messages and a prompt for the names of the subtrees. Enter the name of each subtree as a path name relative to the directory you specified as the last argument of the `dump` command. Do not specify ".." as a subtree or specify subtrees that contain symbolic links. To end the list of subtree names, enter a blank line as shown in the following example, or press Ctrl-D.

```
toaster> dump 0ufnl nrst0a dir.1.3.5 /vol/vol0/home
DUMP: creating "snapshot_for_dump.0" snapshot.
creating.....................................................
DUMP: Date of this level 0 dump: Tue Jun  3 12:47:14 1997
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /vol/vol0/home to nrst0a
DUMP: mapping (Pass I) [regular files]
DUMP: Reading file names from standard input
dir1
dir3
dir5

DUMP: 96% done, finished in 0:07
DUMP: 3252426 tape blocks
DUMP IS DONE
DUMP: deleting "snapshot_for_dump.0" snapshot.
deleting.....................................................
```



CAUTION: When you do a level 0 dump for multiple subtrees, make a note of the subtrees you are dumping and the dump name. For subsequent incremental dumps, always specify the same dump name and the same subtrees. The `dump` command does not enforce the correspondence between the subtree names and the dump name. Therefore, you *must* keep track of this information yourself. If you do not specify the same subtrees for the incremental dumps, you cannot do incremental restores.

Restoring dump files

You can restore a dump file created by the filer `dump` command in one of two ways:

- Using the `restore` command on the filer to restore the file from a local or remote tape drive. If you use a remote tape, the host for the tape drive must support the `rmt` protocol.
- Using the UNIX `ufsrestore` command on a client that supports the SunOS 5.*x* / Solaris 2.*x* `ufsdump` format.



CAUTION: Use the SunOS `ufsrestore` command only if your filer runs NFS exclusively. If your filer runs the CIFS protocol, do not use the SunOS `ufsrestore` command. Doing so results in data loss.

The `restore` command is usually used in the following situations:

- The entire filer is damaged and unusable. If this happens, call Network Appliance Technical Support to see whether you can repair the filer. If the filer cannot be repaired, you need to reinitialize the disks and restore all file systems from tape. If you can repair the filer but some files are deleted during the repair process, you need only restore the missing files from tape.
- If no disk slots are available for storage expansion, you can back up the entire filer, replace the current disks with disks of greater capacity, set up volumes on the new disks, and restore the filer from tapes.
- If some files on the filer are corrupted, you can restore the subtree containing the files.
- Recover files that have been deleted from the filer but have been backed up to tape.



CAUTION: Before trying to initialize disks on your filer, call Network Appliance Technical Support for instructions. Initializing disks destroys all data on your disks.

Restoring quota trees

Before using the `restore` command, you must re-create quota trees; otherwise, the `restore` command creates ordinary directories in place of the quota trees. Obtain a list of the quota trees on the filer that you backed up, then re-create them.

Generating a list of quota trees

You can generate a complete list of quota trees in an active file system by entering the following command:

```
quota qtree
```

If you do not have access to an active file system, use the record of the */etc/quotas* file you made, as described in "Restricting disk usage by using disk quotas" on page 213.

Re-creating quota trees

You re-create each quota tree by issuing the following command:

```
quota qtree tree
```

where *tree* is a quota tree.

The restore command

This section discusses only the syntax of the `restore` command. Refer to "Locally restoring a subtree" on page 249 or "Restoring the entire filer" on page 251 for more information and for step-by-step procedures for restoring files. For information about using the tape drive on another filer, see "Using a filer tape drive from another system" on page 257.

The filer `restore` command is similar to the one in SunOS, except that the filer `restore` command doesn't support the `i` argument of the Solaris `ufsrestore` command This argument enables you to

specify interactively individual files and directories to be restored from a dump tape. However, you can restore individual files and directories using the x argument of the filer `restore` command.



CAUTION: You can use the filer `restore` command only to restore dump files created by the filer. Do not try to use the filer `restore` command to restore dump files created by other servers or workstations.

You can enter the `restore` command on a trusted host through `rsh`. In addition, you can run multiple `restore` commands by starting each from a remote shell. The filer supports up to two simultaneous restores.

The restore command syntax

Following is syntax of the `restore` command:

```
restore [function_key] [options] [arguments]
[subtree]
```

After the function key (described in Table 11-2), put all the option keys to the `restore` command together with no intervening spaces, then put the arguments to each option in the order of the options, and end with the subtree parameter.

The `r`, `R`, `t`, and `x` keys specify the functions of the restore. You can have, at most, one of these in the command. They are explained in Table 11-2.

Table 11-2. Restore command function keys

Key	Meaning
r	Rebuilds the file system or subtree. If you are applying incrementals, this must be the only option.
R	Specifies a particular tape if the dump file is on a multitape-volume set. Use this argument to restart `restore` if it has been interrupted.

Table 11-2. Restore command function keys (continued)

Key	Meaning
t	Lists all the file names in the archive file that you specify with the `f` option. If you don't specify any archive, the root directory of the dump tape is listed.
x	Extracts an individual file or directory from the backup tape.

You can include zero or more of the option keys listed in Table 11-3 after the function key with no intervening spaces and no space after the function key.

Table 11-3. Restore options

Option	argument	Meaning
b	*blocks*	Specifies the block size. Use the same block size you used when dumping the file.
f	*archive*	Specifies the name of the archive file. If you specify "-", `restore` reads from standard input.
D	*pathname*	Specifies the absolute path name of a directory into which the files are restored. Without the path name, the files are restored to the directory from which they were dumped.
h		Prevents hierarchical restoration.
m		Extract by inode number rather than file name.

Table 11-3. Restore options (continued)

Option	argument	Meaning
s	*number*	Specifies the number of the file if multiple dump files exist on a tape. File numbering starts at 1.
v		Specifies that `restore` takes place in verbose mode. That is, `restore` displays each file name preceded with its file type. The filer can restore files faster without the `v` option.
y		Specifies that `restore` not ask whether it should abort when getting an error. That is, if there are bad blocks, `restore` skips over them and continues. This option is particularly useful if you use `restore` through `rsh`. This is because without the `y` option, if `restore` through `rsh` encounters some read error, it aborts immediately.

Locally restoring a subtree

This section provides an example illustrating how you can restore a subtree by the `restore` command on the filer. The example assumes the following:

- The name of the filer is toaster.
- The subtree to be restored is */vol/vol0/etc/home*.
- */vol/vol0/etc/home* was the backed up subtree; it is not a directory within a subtree.

Following is the procedure for restoring */vol/vol0/etc/home*:

1. Delete all files in the subtree before the restore.

2. Install the tape that contains the level 0 dump file for the *home* subtree in a local or remote tape drive.

   If you restore from a remote host, make sure that the filer and the remote host with the tape drive (the tape host) can perform host name lookups. If either host relies on its */etc/hosts* file, that file must have an entry for the other host.



NOTE: The `restore` command doesn't support incremental-only restores. You must begin the restoration from a level 0 dump.

3. If you are restoring from a local tape, enter the `sysconfig -t` command on the filer to determine the name of the tape device. If you are restoring from a remote tape, specify the remote tape in the following format:

   hostname:*device_name*

   This procedure uses the following tape device name:

   `unixhost:/dev/rst0a`

4. Enter the following command if the subtree is dumped to only one tape:

   `toaster>` **`restore rfD rst0a /vol/vol0/etc/home`**

   It is not necessary to specify the `D` option and */vol/vol0/etc/home* if the destination directory for the restore is the same as the directory that was backed up. This command restores the dump file in *rst0a* to the */vol/vol0/etc/home* directory, using the same block size as when the subtree was dumped to the tape. If backup of the subtree is contained in multiple tape volumes, `restore` prompts you for the next tape volume when appropriate.

5. After the restore is finished, remove the tape from the drive. Install another tape that contains the next lowest level of incremental dump. For example, if you have dump tapes containing level 1, level 2, and level 3 dumps of */vol/vol0/etc/home*, load the tape with the level 1 dump in the drive and repeat the `restore` command. Repeat this step until the dump file of the highest level (level 3 in this example) is restored.

6. After the last incremental backup was restored, from a client, remove the *restore_symboltable* file in the directory that you just restored. This file was created by `restore` to pass information between different levels of restore.

The restore process is now complete.

Restoring the entire filer

To restore the entire filer, repeat the procedure described in this section for each volume. This section assumes that you already initialized the disks on your filer.

If each disk volume was backed up to one tape volume

If each disk volume was backed up to one tape volume, follow the procedure in "Locally restoring a subtree" on page 249 to restore each volume. The only difference is that you use */vol/volume_name* as the directory to which the dump file is restored.

If each volume was backed up as subtrees

If each volume was backed up as subtrees, restore each subtree. For example, if the root volume contains two directories, */vol/vol0/etc* and */vol/vol0/home*, and you backed them up as subtrees, follow these steps to restore the entire volume:

1. If you want to restore quota trees, create directories on the filer using the `quota qtree` command that is used as the top of the subtrees to be restored. For example:

   ```
   toaster> quota qtree etc
   toaster> quota qtree home
   ```

2. Install the tape that contains the level 0 dump file for the */vol/vol0/etc* subtree in a local or remote tape drive.

   If you restore from a remote host, make sure that the filer and the remote host with the tape drive (the tape host) can perform host name lookups. If either host relies on its */etc/hosts* file, that file must have an entry for the other host.



NOTE: The `restore` command doesn't support incremental-only restores. You must begin the restoration from a level 0 dump.

3. If you are restoring from a local tape, enter the `sysconfig -t` command on the filer to determine the name of the tape device. If you are restoring from a remote tape, specify the remote tape in the following format:

   hostname:*device_name*

   Following is the sample tape device name used in this procedure:

   `unixhost:/dev/rst0a`

4. Enter the following command on the filer:

   ```
   toaster> restore rfD rst0a /vol/vol0/etc
   ```

This command restores the dump file in rst0a to the */vol/vol0/etc* directory. If the dump file for the subtree is in multiple tape volumes, `restore` prompts you for the next tape volume when appropriate. See "Examples of restoring from multiple tapes" on page 253 for a sample screen display when more than one tape contains the dump volume.

5. After the restore is finished, remove the tape from the drive. Install another tape that contains the next lowest level of incremental dump. For example, if you have dump tapes containing level 1, level 2, and level 3 dumps of */vol/vol0/etc*, load the tape with the level 1 dump in the drive and repeat the `restore` command. Repeat this step until the dump file of the highest dump level (level 3 in this example) is restored.

6. From the client, remove the *restore_symboltable* file in the directory that you just restored. This file was created by `restore` to pass information about different levels of restore.

7. Repeat Steps 4 through 6 for the */vol/vol0/home* subtree.

If the filer contains multiple volumes, repeat the preceding procedure for each volume.



NOTE: If your filer currently contains multiple volumes but the system backup was made before multiple volumes were installed, specify the path name into which the dump file is to be restored. For example, when the filer contained a single volume, you backed up the */home* directory. Now you want to restore */home* from tape to */vol/engineering/home*. Specify */vol/engineering/home* as the target directory in the `restore` command. If you don't specify the directory, the dump file *home* is restored to */vol/vol0/home*.

Examples of restoring from multiple tapes

This section provides additional examples illustrating how to restore a volume or a subtree that was backed up to multiple tapes.

Restoring a volume from multiple tapes to a directory

In this example, you have backed up the *vol0* volume to two tapes using the following command:

```
toaster> dump 0fB rst0a,rst1a 600 /vol/vol0
DUMP: creating "snapshot_for_dump.1" snapshot.
DUMP: Date of this level 0 dump: Wed Jan 21 20:28:07
1998
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /vol/vol0/ to rst0a,rst1a
DUMP: mapping (Pass I)[regular files]
DUMP: mapping (Pass II)[directories]
DUMP: estimated 857 tape blocks on 1.40 tapes
DUMP: Dumping volume 1 on rst0a
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: Dumping volume 2 on rst1a
DUMP: Volume 2 begins with blocks from inode 2880
DUMP: 821 tape blocks
DUMP IS DONE
DUMP: deleting "snapshot_for_dump.1" snapshot.
```

At the time of the restore, the filer prompts you for the next tape device. The following example restores the volume into a directory named */vol/vol0/myexample*, and the two tapes are in different tape drives:

```
toaster> restore rfD rst0a /vol/vol0/myexample
Mount tape volume 2
Enter ''none'' if there are no more tapes
otherwise enter tape name (default: rst0a) rst1a
```

If the two tapes use the same tape device for example, `rst0a`, remove the tape currently in the tape drive and load the next tape. Then accept the default tape device name (in this example, `rst0a`) when the filer prompts you for the name of the device for the second tape.

Restoring a named file from multiple tapes

If you want to restore a specific file or directory from a subtree, use the `x` option in the `restore` command. Specify the path name of the file or directory relative to the subtree that was backed up. For example, if the *vol0* volume was backed up as a subtree and you want to restore all contents of the */vol/vol0/test* directory, specify */test* as the path name in the `restore` command.

In the following example, the subtree containing the directory to be restored was backed up to two tapes:

```
toaster> restore xf rst0a /test
```

In this command, you must specify the tape device for the first tape (it is `rst0a` in this example). This is necessary because `restore` needs to read information about the directory structure of the subtree from the first tape before restoring the data.

The filer then displays the following messages to let you specify the tape devices containing the multiple tape volumes. When the filer prompts you for the tape volume number, start with the last tape volume, as shown in this example:

```
You have not read any tapes yet.
Unless you know which volume your file(s) are on you
should start with the last volume and work towards the
first.
Specify next volume #: 2
Mount tape volume 2
Enter ''none'' if there are no more tapes
otherwise enter tape name (default: rst0a) rst1a
You have read volumes: 2
Specify next volume #: 1
Mount tape volume 1
Enter ''none'' if there are no more tapes
otherwise enter tape name (default: rst1a) rst0a
```

After prompting for the tape devices, the filer also displays the following question:

```
set owner/mode for '.'? [yn] y
```

To keep the original owner and permission modes for the restored files and directories, enter **y**.

Using a filer tape drive from another system

You can use a filer tape drive to dump and restore files on a UNIX system or on another filer. The filer with the tape drive must allow access from the other system. That is, the host or filer from which a `dump` (or `ufsdump`) or `restore` (or `ufsrestore`) command is issued must be listed in the following configuration files:

- the */etc/hosts.equiv* file on the filer with the tape drive. Alternatively, you can specify both the host and user in */etc/hosts.equiv*. The filer */etc/hosts.equiv* file contains entries in this format:

  ```
  hostname [ username ]
  ```

  For more information about */etc/hosts.equiv*, see the *na_hosts.equiv*(5) man page.

- the */etc/hosts* file on the filer with the tape drive or in the DNS database if the filer is using DNS.

Also, the filer with the tape drive must be added to the other system's */etc/hosts* file. See the *Quick Start Guide* for information about how to add the filer to the client's */etc/hosts* file.

In the commands for dumping and restoring, specify the filer tape drive in the following format:

`filer``:`*`device_name`*

The following examples show how to use a filer tape drive from an SVR4 UNIX system such as Solaris 2.*x*:

- To dump a file system that resides on a UNIX system

  ```
  client% ufsdump 0uf toaster:nrst0a /dev/sd0g
  ```

- To restore a directory from a filer tape drive to a UNIX system

  ```
  client% cd /filesystem
  client% ufsrestore rf toaster:rst0a
  ```

The following examples show how to use a filer's tape drive from a UNIX system other than an SVR4 UNIX system:

- To dump a file system that resides on a UNIX system

```
client% dump 0uf toaster:nrst0a /dev/sd0g
```

- To restore a directory from a filer tape drive to a UNIX system

```
client% cd /filesystem
client% restore rf toaster:rst0a
```

To use a filer tape drive from another filer, follow the directions in "Dumping multiple subtrees" on page 243, "Locally restoring a subtree" on page 249, and "Restoring the entire filer" on page 251 to dump or restore files, using *filer*:*device_name* as the tape drive name in these procedures.

Backing up the filer using NDMP

You can use third-party backup management programs that support NDMP, Version 1 or 2, to back up the filer. The filer can be backed up by NDMP to the following devices:

- tape drives, libraries, or stackers attached to the filer
- tape drives, libraries, or stackers attached to the workstation that runs the backup program
- tape drives, libraries, or stackers attached to a workstation or filer on your network

Figure 11-1 shows an example of backing up the filer to various devices using NDMP.

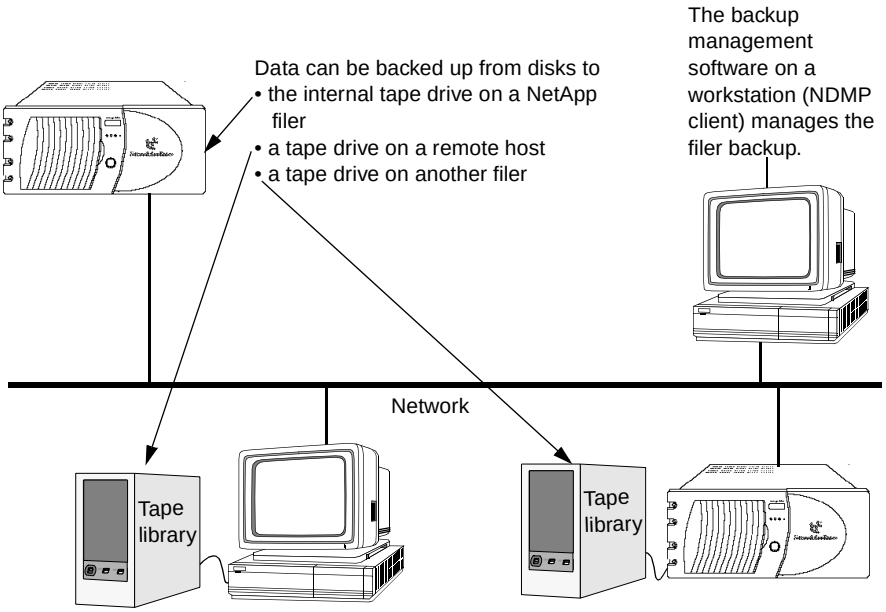


Figure 11-1. Using NDMP to back up data to tape

Use the ndmpd command for enabling NDMP support on the filer and for obtaining information about backup and restore operations that use NDMP. Following is the syntax for the ndmpd command:

```
ndmpd [on|off|status|probe [session]|kill [session]]
```

The on option enables NDMP. The off option disables NDMP. After you enter ndmpd off, the filer continues processing existing requests but rejects new ones.

If you want the filer to run NDMP after each reboot, put the ndmpd on command in the */etc/rc* file.

The status option displays the current state of each of the five possible NDMP sessions. Sessions are numbered 0 to 4. In the following example, there is only one active session:

```
toaster> ndmpd status
ndmpd ON.
Session: 0
  Active
  tape device: open
  state:      Idle
Session: 1
  Inactive
Session: 2
  Inactive
Session: 3
  Inactive
Session: 4
  Inactive
```

The probe option displays detailed information about the NDMP session specified. Using the probe option without a session parameter displays NDMP information for all five sessions. In the following example, there is only one active session:

```
toaster> ndmpd probe 0
Session: 0
  pid:                    0xfffffc00005d1e10
  is_active:              TRUE
  eof:                    FALSE
  authorized:             TRUE
  my_sequence:            32
  tape.id:                16973824
  tape.device:            nrst0a
  tape.mode:              Read/Write
  protocol_version:       1
  record_size:            61440
  data.bu_type            nil
  data.state:             Idle
  data.operation:         None
  data.halt_reason:       N/A
  data.pause_reason:      N/A
  data.seek_position:     0*0
  data.blocks_processed:  0
  data.est_blocks_remain: 0
  data.est_time_remain:   0
  data.window_begin_byte: 0*0
  data.abort:             FALSE
```

The `ndmpd kill` *session* command stops all processing in the specified session and puts the session in an inactive state. Use this command if you want to clear a hung NDMP session.

Controlling tape devices

You can control tape devices with the `mt` command. This section describes how to do the following:

- move a tape to the next file (fsf)
- append a dump on a tape by moving to the end of data (eom)
- rewind a tape
- take a tape off-line
- display status information about a tape device

The examples in this section use the device nrst0a—a no-rewind tape device, unit zero, at the highest format (density).

The mt command syntax

The syntax of the `mt` command is as follows:

`mt [-f|-t]` *`tapedevice command`* `[` *`count`* `]`

This section discusses only the `eom`, `fsf`, `rewind`, `offline`, and `status` commands. Keep in mind the following information:

- The `-f` and `-t` options are interchangeable as far as the filer is concerned. Only the `-f` option is shown here.
- For additional information about controlling tape devices and detailed information about the `mt` command, consult the *mt*(1) man page.
- For information about the format of a filer tape device name, consult the *tape*(4) man page.

Moving a tape to the end of data

You can append material, such as a dump, on a tape device. To do so, use the `eom` command, which moves the tape to end of data (end of media if tape is full), as in the following example:

```
toaster> mt -f nrst0a eom
```

After this command, you can write to the remainder of the tape. Make sure that there is enough tape for your additional data.

Appending a dump

To skip over a previously created dump file and append a dump on a tape device, use the `fsf` command, as in the following example:

```
toaster> mt -f nrst0a fsf 1
```

Rewinding a tape

To rewind a tape, use the `rewind` command, as in the following example:

```
toaster> mt -f nrst0a rewind
```

Taking a tape drive off-line

To rewind the tape and, if appropriate, take the tape drive off-line by unloading the tape, use the `offline` command, as in the following example:

```
toaster> mt -f nrst0a offline
```

Displaying status information

To display status information about the tape unit, use the `status` command, as in the following example:

```
toaster> mt -f nrst0a status
Tape drive: Quantum DLT7000
Status: ready, write enabled
Format: 85937 bpi 70 GB (w/comp)
fileno = 0  blockno = 95603  resid = 0
```

Chapter 12: System Information and Performance

This chapter describes the following topics:

- how to use the `version`, `sysconfig`, `vol status`, `sysstat`, `uptime`, and `netstat` commands to display information about system configuration and statistics
- how to improve filer performance by adding hardware or setting software options

Displaying the Data ONTAP version

To display the version of Data ONTAP currently running on a filer, use the `version` command. The display shows the version number and the date of the version, as follows:

```
toaster> version
NetApp Release 4.3: Fri Sep 12 03:06:00 PDT 1997
```

Displaying filer configuration information

The *sysconfig* command displays information about the filer's hardware configuration. The exact types of information displayed depend on the command options.

Displaying disk information using sysconfig -d

The `sysconfig -d` command displays product information about each disk in the filer.

Displaying RAID information using sysconfig -r

The `sysconfig -r` command displays RAID configuration information about the parity disk, data disks, and hot spare disks (if any). This information is useful for the following purposes:

- Locating a disk referenced in a screen message. Refer to "Using disks of various sizes" on page 61 for more information about disk identifiers.
- Determining how much space on each disk is available to the filer. Refer to "Usable space on each disk" on page 62 for more information about disk capacity.
- Determining the status of the disk operations such as RAID scrubbing, reconstruction, parity verification, adding a hot spare, and disk failure.

You can obtain the information displayed by `sysconfig -r` from SNMP, using the Network Appliance custom MIB. For information about SNMP, see "Using SNMP" on page 80.

Displaying tape drive information using sysconfig -t

The `sysconfig -t` command displays device and configuration information for each tape drive on the system. Use this command to determine the capacity of the tape drive and the device name before you use the `dump` and `restore` commands.

Displaying overall filer information using sysconfig -v

The `sysconfig -v` command displays the system's RAM size, NVRAM size, and information about devices in all expansion slots. This information varies according to the devices on the filer. You can specify a slot number to display information about a particular slot. Slot numbers start at 0, where slot 0 is the system board.

Displaying overall filer information using sysconfig

If you enter **`sysconfig`** without any options, information similar to what you get with `sysconfig -v` is displayed, but the information is abbreviated. When you report a problem to Network Appliance, provide the information displayed by `sysconfig -v`. This information is useful for diagnosing system problems.

Displaying volume information

The `vol status` command displays information about a volumes's configuration. The exact types of information displayed depend on the command options. When you specify a volume, the information for that volume is displayed; when you do not specify a volume, the status of all volumes in the filer is displayed.

Displaying volume state information with vol status

With no options, the `vol status` command displays a one-line synopsis of volumes. This includes the volume name, whether it is on-line or off-line, other states (for example, partial, degraded, and so on).

Displaying disk information using vol status -d

The `vol status -d` command displays information about disks. The disk information is the same as the information from the `sysconfig -d` command.

Displaying RAID information using vol status -r

The `vol status -r` command displays a list of the RAID information. The display is the same as the `sysconfig -r` display.

Displaying RAID information for each group using vol status -v

The `vol status` -v command displays information about each RAID group.

Displaying filer statistics

You use the `sysstat` and `uptime` commands to display filer statistics.

The sysstat command

The `sysstat` command displays information about CPU utilization, file operations, read and write operations on disks and tapes, and the age of data in the cache buffer. By default, the filer displays statistics every 15 seconds; you can specify the interval, in seconds, at which statistics are displayed.

The `sysstat` output is particularly useful for revealing file access patterns on your filer, from which you can determine whether you should install more NVRAM, system memory, or disks.

Depending on the applications that access the filer, the computers on the network, and the network configuration, the following suggested actions might improve your filer's performance:

- Typically, 2 MB of NVRAM is sufficient for a system with one SCSI channel. However, if the Disk Kb/s write column shows that the filer writes to disks almost every second, adding NVRAM might help improve system performance. If there are short spurts of writes, adding NVRAM does not improve system throughput. See the hardware guide for your filer model for information about how much additional NVRAM you can install.

- The data in the Cache age column indicates how fast read operations are cycling through system memory. If the values in the Cache age column are consistently below 5, the filer might benefit from more system memory. Cache age shows the age of the oldest read-only blocks in memory. A low cache age means that the filer is retrieving information from disk instead of from memory. For information about adding memory, see the next section, "The uptime command."

- If the CPU utilization percentage in the CPU column is low but you are not getting the expected performance, you might need to add disks to the system.
- If CPU utilization is high, you might need to add another filer.
- If the network traffic shown in the Net Kb/s column is at or near the capacity of the network interfaces (1 MB per Ethernet adapter and 10 MB per FDDI adapter), you might need to add another network adapter card to the system.

The uptime command

The `uptime` command prints the current time, the length of time the system has been up, and the total number of NFS operations the system has performed since it was last booted. A example of the display is

```
toaster> uptime
8:54am up 2 days 22:23, 3122520 NFS ups
```

Improving filer performance

The following list describes configuration procedures that might improve your filer's performance:

- Limiting directory file size

  An extremely large directory file can use up most of the filer's CPU cycles when a user enters an `ls` command in the directory. The limit on directory file size is set by the `options wafl.maxdirsize` command. The default limit (10 MB) should prevent the system from hanging. A directory of this size accommodates approximately 300,000 files with short file names.

  The `wafl.maxdirsize` option takes the maximum number of kilobytes as its argument. When you reset the maximum directory size, the argument that you supply is rounded up to the next highest 4K boundary. If a user tries to create a file that would cause the directory to grow larger than the specified size limit, the user's command fails.

- Balancing NFS traffic on network interfaces

  To balance network traffic among different interfaces, attach multiple interfaces on the filer to the same physical network. For example, if two Ethernet interfaces on the filer named toaster are attached to the same network where four clients reside, specify in */etc/fstab* on client1 and client2 that these clients mount from *toaster-0:/home*, and specify in */etc/fstab* on client3 and client4 that these clients mount from *toaster-1:/home*. This scheme can balance the traffic among interfaces if all clients generate about the same amount of traffic.

  The filer always responds to an NFS request by sending its reply on the interface on which the request was received.

  For information about editing */etc/fstab*, see the *Quick Start Guide*.

- Avoiding access time update for inodes

  If your applications do not depend on having the correct access time for files, you can disable the update of access time (atime) on an inode when a file is read. To prevent updates, turn on the `no_atime_update` option. Consider turning on this option if your filer has extremely high read traffic (for example, on a news server used by an internet provider), because it prevents inode updates from contending with reads from other files.



CAUTION: If you are not sure whether your filer should maintain an accurate access time on inodes, leave this option at its default (off) so that the access time is updated.

- Putting the file ID in the file handles reported by the filer in "big-endian" format (applicable to NFS only).

  Turning on the `nfs.big_endianize_fileid` option improves performance on directory lookups for clients that use the file ID in the file handle as a hash key in certain ways. Enable this option if your NFS clients are mainly running HP-UX or IRIX.

NOTE: If you turn on the `big_endianize_fileid` option, all NFS clients that have mounted directories from the filer must unmount and remount them; otherwise, they get "stale file handle" errors on all references to files already opened on the filer until they unmount and remount all directories.

- Enabling minimal read-ahead.

  If the file access patterns of your clients are random (nonsequential), turning on minimal read-ahead might improve performance. By default, the filer uses aggressive read-ahead, which enhances sequential access (more commonly used by UNIX clients and applications). To specify minimal read-ahead, turn on the `minra` option. By default, the option is off and the filer does very aggressive read-ahead.

Displaying network statistics using netstat

The `netstat` command displays network-related data in various output formats.

The `-i` and `-I` options show the state of all network interfaces or one specific interface, respectively. The `netstat -r` command shows the filer's routing table.

For information about troubleshooting network problems, see "Network problems" on page 292. For more information about the `netstat` command, see the *na_netstat*(1) man page.

Chapter 13: **Troubleshooting**

This chapter describes the following topics:

- getting technical assistance
- booting from the system boot diskette
- restarting a down filer
- NVRAM problems
- volume problems
- disk problems
- configuration problems
- network problems
- NFS problems
- UNIX `cpio` problems
- UNIX `df` problems
- Panic messages

Hardware problems are described in the hardware guide for your filer model. See the appropriate hardware guide for information about system LEDs, LCD error messages (if your filer has an LCD), and system startup messages.

Getting technical assistance

If you encounter problems with the filer that you cannot solve, you might need to contact your service provider or Network Appliance Technical Support for assistance. If you do, have the following information available:

- the system serial number
- the system configuration as reported by the `sysconfig -v` command (if the filer still responds to the command)
- any diagnostic messages that were reported (Diagnostic messages are in the */etc/messages* file on the root volume.)

Network Appliance provides telephone, FAX, and email support. You can contact them by telephone, FAX, or email, as listed in Figure 13-1.

Figure 13-1. Contacting Network Appliance Technical Support

Telephone	FAX	Email
For U.S. customers: 888 463-8277 For international customers: +1 408 367-3000 (ask for Technical Support)	(408) 367-3151	support@netapp.com

Booting from diskette

You might need to reboot the filer from the system boot diskette to correct configuration problems, recover from a lost password, or correct certain disk configuration problems.

NOTE: Do not try to boot the filer with a pre-5.0 system boot diskette. The attempt to boot fails because pre-5.0 software cannot recognize system data from a filer running Release 5.0 or later. If you need to run an earlier release on the filer, contact Network Appliance Technical Support, as described in "Getting technical assistance" on page 277, for instructions about how to downgrade your filer.

To boot from diskette, display the boot menu, then choose a type of boot.

Displaying the boot menu

To display the boot menu from a diskette, follow these steps:

1. Place the system boot diskette into the disk drive.
2. Reboot the system using one of these methods:
   - If the console is showing the `mon>` prompt, enter **`b`** (for boot). If the console is showing the `ok` prompt, enter **`boot`**.
   - If the console is unresponsive, reset the filer by turning it off, leaving it off for 30 seconds, and turning it back on.

   Booting from the diskette takes longer than booting from the hard disk. During the boot process, the filer displays the boot menu, shown in Figure 13-2.

```
1) Normal Boot
2) Boot without /etc/rc
3) Change Password
4) Initialize all disks
5) Maintenance mode boot

Selection (1-5)?
```

Figure 13-2. The boot menu

Choosing a boot type

To choose a boot type, enter the number next to the boot type.

There are three boot types, as shown in Table 13-1.

Table 13-1. Boot types

Boot type	Use
Normal boot	Use normal boot to run the filer normally, but from a diskette.
Boot without */etc/rc*	Boot without */etc/rc* to troubleshoot and repair configuration problems.
Maintenance mode boot	Use maintenance mode boot to go into maintenance mode and perform some volume and disk operations and get detailed volume and disk information. Maintenance mode is special for the following reasons: • Most normal functions, including file-system operations, are disabled. • A limited set of commands is available for diagnosing and repairing disk and volume problems. • You exit maintenance mode with the `halt` command.

Restarting a shutdown filer

The following procedure describes what to do if your filer unexpectedly shuts down:

1. Write down the messages displayed on the console and the message on the LCD (if your filer has an LCD).

2. Perform the actions listed in the following table, depending on the console message:.

If the console is	Do this
Showing the `mon>` prompt	Enter **`b`** (for boot) to reboot the filer.
Showing the `ok` prompt	Enter **`boot`** to reboot the filer.
Not responding	Reset the filer by turning it off, leaving it off for 30 seconds, and turning it back on.

3. If the filer still does not boot, display the boot menu from diskette, as described in "Booting from diskette" on page 278, then choose Normal boot.

NVRAM problems

This section describes two types of NVRAM problems: NVRAM battery failure and inconsistent NVRAM contents.

NVRAM battery failure

This section applies only to the NetApp F220, NetApp F330, and NetApp F540 filers. If the filer detects low batteries on the NVRAM card, it gives advance warning and automatically shuts down every 24 hours (you can change the time interval with the `raid.timeout` option to the `options` command).

If the filer is in degraded mode and the NVRAM batteries are low, it fails to boot. Replace the NVRAM batteries immediately.

Very rarely do NVRAM battery failures cause data loss. When the filer executes the `halt` command, it flushes the contents of NVRAM to disk and turns off the NVRAM so that no battery power is used.

Inconsistent NVRAM contents

The filer performs a number of checks to ensure that the NVRAM contents are consistent. If the contents are inconsistent, the filer performs one of following actions:

- If the inconsistency is due to a failure to update a volume properly, the filer displays a message suggesting that you halt the filer with the `halt` command, take the offending volume off-line in maintenance mode, and reboot the filer.
- If the inconsistency is due to the log having updates for an off-line volume, the filer asks whether to discard them.
- If there are many inconsistencies that cannot be repaired, the filer discards the inconsistent contents and dumps core. The file requests received during the last few seconds before the filer shuts down are lost. This does not cause the file system to become inconsistent, but files written during the last 10 seconds before shutdown might contain old or incorrect data. Also, because the parity of some recently written stripes might

be incorrect, the filer must do a parity check on the entire RAID array. The parity check (and any correction) is performed on-line; that is, the filer conducts the test in the background while continuing otherwise normal operation.

This kind of data error can happen only when you boot a system after a failure or after you turn off a filer without using the `halt` command.

Volume problems

The following sections discuss some volume problems.

Failed mounts and stale file handles

If mounts fail and clients see stale file handles, it might be because you renamed a volume and it is not exported anymore. This is because the new volume name was not in the */etc/exports* file on the root volume. Edit the */etc/exports* file on the root volume to change the old volume name to the new volume name.

Volume name problems

A valid volume name has the following characteristics:

- It begins with the prefix /vol/.
- The prefix is followed by either a letter or an underscore ("_").
- It contains only letters, digits, and underscores.
- It is not longer than 255 characters.

Examples of valid volume names are

- */vol/_tech_pubs*
- */vol/SW_Engineering*
- */vol/Dept_32*

You might get a message that contains one of the following phrases:

- invalid volume name
- unrecognized volume name
- illegal volume name

If you get one of these messages, do one of the following actions:

- Type correctly the name of an existing volume.
- Type a valid volume name.

Nonexistent disks

If you get a message indicating that a disk does not exist, especially if you are adding a disk to a volume or a RAID group, make sure that the following conditions are in effect:

- The disk is specified correctly.
- The specified disk is a spare.

Disk in use

If you are adding a disk to a volume and you get a message indicating that a disk is in use, the disk you specified might already be a system disk. Make sure that the following conditions are in effect:

- The disk is specified correctly.
- The specified disk is a spare.

System doesn't come up because of missing drives

You might get a message similar to the following:

```
The system cannot boot with more than one disk missing
from a RAID group.
```

This message indicates that a volume might be missing some drives because either not all of the drives in a volume were transferred to a new filer or disks were damaged.

Make sure that all the disks in a volume were transferred. If the problem persists, it might mean that disks were damaged. Follow these steps:

1. Boot in maintenance mode.
2. Use the `vol offline` command to take the volume off-line.
3. Reboot the filer; the volume specified in Step 1 is off-line.
4. Add missing disks if possible, then bring them on-line; otherwise, follow these steps:
   a. Replace any broken disks.
   b. Destroy the old volume.
   c. Create a new volume.
   d. Use the `restore` command to restore the contents of the old volume from a backup tape.

Disk problems

How disk failures affect the filer depends on whether a hot spare disk is installed.

Disk failure without a hot spare disk

If a hot spare disk is not installed, a disk failure causes automatic shutdown after a specified period.

Single disk failure

If a disk fails and there are no hot spare disks to replace it, the filer continues to run without losing any data but has a somewhat degraded performance. Replace the disk as soon as possible, because a second disk failure could cause the filer to lose the entire file system.

The filer logs a warning message in the */etc/messages* file on the root volume every hour after a disk fails. To ensure that you notice the failure, the filer automatically shuts itself off in 24 hours, by default, or in a period set with the `raid.timeout` option to the `options` command. You can restart the filer without fixing the disk, but it continues to shut itself off periodically until you repair the problem.

Check the */etc/messages* file on the root volume once a day for important messages. Checking this file can be automated with a script that periodically searches the file and then alerts you.

On-line reconstruction

After you replace a disk, the filer detects the new disk when the system boots. The filer starts file service and reconstructs the missing data in the background with minimum interruption to service.

Disk failure with a hot spare disk

If a hot spare disk is available, when the filer detects a failure, it replaces the failed disk with a hot spare disk. The missing data is reconstructed on the hot spare disk in the background, so that the interruption to file service is minimized. The activity is logged in

the */etc/messages* file on the root volume, but there is no automatic shutdown. After the filer is finished reconstructing data on the hot spare disk, replace the failed disk with a new hot spare disk as soon as possible so that there is always a hot spare disk available in the system. For information about replacing a disk, refer to "Disk management tasks" on page 71.

If a second disk fails and there is no hot spare disk available, contact Network Appliance Technical Support, as described in "Getting technical assistance" on page 277.

In addition to disk failure and hot spare disk replacement activity, the */etc/messages* file on the root volume logs any failure in a periodic check of the hot spare disk. For more information, refer to the *na_sysconfig*(1) and *na_messages*(5) man pages.

Inconsistent file system

Usually, the file system never becomes inconsistent. However, an inconsistent file system can be a result of combined disk and NVRAM failure. If your file system becomes inconsistent, contact Network Appliance Technical Support for assistance, as described in "Getting technical assistance" on page 277

Disk operations in maintenance mode

For information about maintenance mode, see "Choosing a boot type" on page 279.

You can perform the following operations to troubleshoot disk problems in maintenance mode:

- Obtain detailed device information for each disk with the `disk_list` command.
- Check access to a particular disk with the `disk_check` command.
- Erase a disk label with the `disk_erase_label` command.

Displaying detailed disk information

The `disk_list` command displays detailed device information for each disk on the system, such as its SCSI adapter, SCSI ID, drive type, and firmware revision level.

Checking access to a disk

The `disk_check` command checks access to a particular disk, issuing read requests to the disk. The disk is active for approximately 15 seconds. During this time, you can watch the disk's activity LED to verify that the disk is accessed.

Erasing a disk label

The `disk_erase_label` command erases a disk label on the specified disk drive. Use the `disk_erase_label` command carefully because after a disk label is erased, RAID treats the disk as uninitialized and no longer recognizes it as a member of a RAID array. Typically, this command is used to remove an old RAID label on a previously used disk that is now being added to an existing RAID group. For example, use this command to erase the disk label when you move a disk from one filer to another.

Configuration problems

Configuration problems usually occur in one of the three configuration files on the root volume: */etc/rc*, */etc/exports*, or */etc/hosts*. This section describes common configuration problems.

Filer not accessible from the administration host

If you can access the filer from the console but not from the administration host, the filer's */etc/hosts* file on the root volume might have an IP address that is unreachable. Follow these steps to fix this problem:

1. Log in to the filer from the system console.
2. At the filer prompt, enter the following commands:

```
toaster> ifconfig if mediatype type IP_address netmask netmask
toaster> exportfs -i -o root=adminhost_IP_addr /
toaster> nfs on
```

   In the actual commands, use the interface name appropriate to your filer instead of *if*, *netmask*, and *adminhost_IP_addr*.

3. Mount the root file system from the administration host.
4. Edit the configuration files as described in Chapter 2, "Routine Filer Administration."

Damaged or missing /etc/rc file

If the */etc/rc* file on the root volume is accidentally deleted, the filer automatically runs `setup` the next time it is booted. If the system does not respond to network requests after a boot, check the console to make sure that the system is not waiting for your input.

If the filer cannot boot from the hard disk because of damaged configuration files, you can boot it from the system boot diskette. In this case, you must manually initialize the filer and correct the configuration. See "Booting from diskette" on page 278 for information about booting from the system boot diskette.

If you are running NFS only, you can use the following procedure for recovering from configuration errors in */etc/rc:*

1. Display the boot menu from diskette as described in "Booting from diskette" on page 278.
2. Enter **2** to choose Boot without */etc/rc*.



NOTE: Booting the filer without the */etc/rc* file on the root volume automatically disables CIFS service. You cannot use this procedure using CIFS.

3. At the filer prompt, enter the following two commands:

```
toaster> ifconfig if mediatype type IP_address netmask netmask
toaster> exportfs -i -o \
toaster> access=adminhost_IP_addr,root=adminhost_IP_addr nfs on
```

In the actual commands, use the interface name appropriate to your filer instead of *if*, *netmask*, and *adminhost_IP_addr*.

4. Mount the root file system from the administration host.
5. Edit the configuration files, as described in the *Quick Start Guide*.
6. Remove the system boot diskette from the disk drive and reboot the filer to test the new configuration files.

Although you can correct network problems using various keyboard commands, correct the */etc/rc* file on the root volume so that it initializes the system correctly. Having a tested and working */etc/rc* file ensures that the system can reboot itself if there is a power outage or system software failure.

Lost password

If you forget your filer password, reset the password by using the system boot diskette. To avoid security problems, take care to limit access to the system boot diskette. The following procedure resets the filer password:

1. If the console displays the filer prompt (for example, `toaster>`), place the system boot diskette into the diskette drive of the filer and enter **`reboot`**. Otherwise, reboot from diskette as described in "Booting from diskette" on page 278.

2. When the boot menu appears, enter 3.

3. When the filer prompts you for a new password, enter it at the prompt.

   The system prints the following message:

   ```
   Password Changed
   Hit Return to reboot:
   ```

4. Remove the diskette from the filer's diskette drive and reboot the filer by pressing the Enter key.

Initializing all disks and replacing all file systems

In the menu displayed after you boot the filer from the system boot diskette, option 4, "Initialize all disks," replaces all existing file systems with a single new one.

Network problems

You can detect network problems by going to the filer console and using the `ping` command. The `ping` command checks whether the filer can communicate with other hosts on the network and that other hosts can communicate with the filer. If the filer should be able to connect with a host but `ping` does not respond with the message indicating the host is alive, follow these steps:

1. Check that the network cable is tightly connected to the proper interface connector.
2. Use the `ifconfig` command to verify that the IP address and netmask are set correctly and that the up and running flags are displayed.
3. Use the `arp -a` command to confirm that the filer has the correct IP-to-Ethernet address map for the host you are trying to reach.
4. Use the `netstat -r` command to examine the routing tables.
5. Use the `netstat -i` command to check for excessive errors on the interfaces. If you see excessive input errors (ierrs) or output errors (oerrs), check the network connections on both ends of the connection. Bad transceivers or network hubs can sometimes introduce errors into the network. Collisions reported by `netstat` are a concern only if the filer detects a substantial percentage of collisions as compared to the total packet throughput. The goal is to keep collisions below 5 percent, but a network can operate properly (but slowly) with collision rates as high as 30 percent.
6. Run `routed status` to find out the status of the default router.



NOTE: An improperly set up network router can also cause network problems. If a router is not working correctly or not configured with the filer's address, clients or hosts on the other side of the router cannot access the filer through that router.

7. If you are using the filer on a CIFS network and you are experiencing difficulty accessing the filer using name-based IP operations (for example, ping toaster), create static mappings on your WINS servers for each of the filer's interfaces.

If you encounter other problems, contact Network Appliance Technical Support for assistance, as described in "Getting technical assistance" on page 277.

NFS problems

NFS problems are indicated when the filer and the client can communicate with each other using the `ping` command, and the client can connect to the filer using Telnet, but the client cannot mount files from the filer. When NFS problems occur, follow these steps:

1. Make sure that the filer is licensed for NFS by entering **`license`** at the filer prompt.

   If the following message appears

   ```
   nfs      not licensed
   ```

   or a message with protocols other than NFS appears but NFS is absent, you must get a license for NFS. For information about how to do so, contact Network Appliance Technical Support, as described in "Getting technical assistance" on page 277.

2. Make sure that the filer can correctly look up the client host name.

3. Make sure that NFS service has been turned on using the `nfs on` command.

4. Make sure that the filer and the client are using correct IP addresses and names.

Sometimes a client can see the filer but gets a Permission Denied message when requesting a mount. If this happens, follow these steps:

1. Make sure that you defined the file systems correctly in the filer's */etc/exports* file on the root volume and that you ran the `exportfs` command on the filer.

2. On certain clients, the mount request does not come from the root user using a privileged port. The filer denies such mount

requests by default to ensure secure access. To grant such mount requests, enter the following `options` command:

```
toaster> options nfs.mount.rootonly off
```

To make this change permanent, add the preceding command to your */etc/rc* file on the root volume.

UNIX cpio problems

If you copy large amounts of data using UNIX `cpio` utility, some files might be copied incorrectly. This happens in UNIX versions of `cpio` that still use a 16-bit inode definition number. Large file systems require a 32-bit inode definition number. The problem generally occurs only on file systems with a large number of hard links.

Some versions of `cpio` work by copying each file with hard links once and then re-creating the hard links. Trouble occurs because the inode number is assumed to be 16 bits. If another file has a matching low-order 16 bits, an internal number collision occurs, which `cpio` does not recognize. The `cpio` utility then overwrites the first file and creates files that no longer contain the original data.

Check with your UNIX provider to see whether your version of `cpio` has this problem. If you use SunOS 4.x, ask for the Sun patch 100556-01. This patch works around the problem by requiring that the files have the same UID, GID, mode, mod time, inumber, and device before concluding that they are the same file.

UNIX df problems

Some UNIX versions of the `df` command have file system limits considerably smaller than the file system size supported by the filer. This can cause the UNIX `df` command to show an incorrect and useless amount of filer disk space in use or remaining. However, the disk space you installed in your filer is fully available and you can use it.

To avoid a useless display of disk space on a client system that uses NFS version 2, enable the `nfs.v2.df_2gb_lim` option, as described in "Configuring filer options" on page 28. For additional information, refer to the *na_options*(1) man page.



NOTE: Some DOS, Windows, or Macintosh clients might have a similar display problem; in these cases, enable the `nfs.v2.df_2gb_lim` option.

At all times, the `df` command entered on the filer correctly shows the amount of disk space used and remaining.

If a directory in a quota tree is mounted and a client issues a `df` command on something under that mount point, the command shows the smaller of the client's file system limit or the filer disk space. This makes the quota tree look fuller according to the client `df` command than it actually is. The `quota report` command shows the correct usage within that quota tree.

Panic messages

If your filer has a serious problem, it issues a system panic message similar to the following one:

```
PANIC: system hung (NS0)!
Volume: volname
Version: verno
```

where *volname* is the name of the volume and *verno* is the version number.

This message indicates either a problem with the hardware or a bug in the system software. If you get a system panic message, call Network Appliance Technical Support immediately, as described in “Getting technical assistance” on page 277.



NOTE: The panic class of this message is significant; the actual text of the message varies with circumstances. Be sure to write down the panic message and provide it to Network Appliance Technical Support when reporting a filer problem.

Glossary

ACL	Access control list. A list that contains the users' or groups' access rights to each share.
adapter card	A SCSI card, network card, hot swap adapter card, serial adapter card, or VGA adapter that plugs into a filer expansion slot.
administration host	A client computer that is used to manage a filer through `telnet` or `rsh`.
appliance	A device that performs a well-defined function and is easy to install and operate.
ASUS board	One type of system board used in earlier versions of FAServers.
authentication	A security step performed by a domain controller for the filer's domain, or by the filer itself, using its */etc/passwd* file.
autosupport	A filer daemon that triggers email messages from the customer site to Network Appliance or another specified e-mail recipient when there is a potential filer problem.
big-endian	A binary data format for storage and transmission in which the most significant bit (or byte) comes first.

client	A computer that share files on a filer.
console	A terminal that is attached to a filer's serial port and is used to monitor and manage filer operation.
copy-on-write	The technique for creating snapshots without consuming excess disk space.
degraded mode	The operating mode of a filer when a disk is missing from the RAID array or the batteries on the NVRAM card are low.
disk ID number	A number assigned by the filer to each disk when it probes the disks at boot time.
disk shelf	A shelf that contains disk drives and is attached to the filer.
EISA	Extended Industry Standard Architecture. The bus architecture used in FAServers.
EISA Configuration Utility	A utility located on the EISA system diskette for configuring FAServers.
Ethernet adapter	An Ethernet interface card.
expansion card	A SCSI card, NVRAM card, network card, hot swap card, or console card that plugs into a filer expansion slot.
expansion slot	The slots on the system board in which you insert expansion cards.
FAServer	A Network Appliance filer that uses the EISA (Extended Industry Standard Architecture) bus architecture.
FDDI adapter	An FDDI interface card.
FDDI-fiber	An FDDI adapter that supports a fiber-optic cable.
FDDI-TP	An FDDI adapter that supports a twisted-pair cable.
GID	Group identification number.

group	A group of users defined in the filer's */etc/group* file.
hot spare disk	A disk installed in the filer that can be used to substitute for a failed disk. Before the disk failure, the hot spare disk is not part of the RAID disk array.
hot swap	The process of adding, removing, or replacing a disk while the filer is running.
hot swap adapter	An expansion card that makes it possible to add or remove a hard disk with minimal interruption to file system activity.
inode	A data structure containing information about files on a filer and in a UNIX file system.
interrupt switch	A switch on some filer front panels used for debugging purposes.
magic directory	A directory that can be accessed by name but does not show up in a directory listing. The *.snapshot* directories, except for the one at the mount point or at the root of the share, are magic directories.
mail host	The client host responsible for sending automatic email to Network Appliance when certain filer events occur.
maintenance mode	An option when booting a filer from a system boot disk. Maintenance mode provides special commands for troubleshooting hardware and configuration.

MIME	Multipurpose Internet Mail Extensions. A specification that defines the mechanisms for specifying and describing the format of Internet message bodies. An HTTP response containing the MIME Content-Type header allows the HTTP client to invoke the application that is appropriate for the data received.
NDMP	Network Data Management Protocol. A protocol that allows Network Appliance filers to communicate with backup applications, and provides capabilities for controlling the robotics of multiple tape backup devices.
network adapter	An Ethernet, FDDI, or ATM adapter.
NICE board	One type of system board used in a FAServer.
NVRAM cache	Nonvolatile RAM in the filer, used for logging incoming write data and NFS requests. Improves system performance and prevents loss of data in case of a filer or power failure.
NVRAM card	Adapter card that contains the filer's NVRAM cache.
parity disk	Disk on which parity information is stored for the RAID-4 disk drive array. Used to reconstruct data in failed disk blocks or on a failed disk.
PCI	Peripheral Component Interconnect. The bus architecture used in newer filer models.
pcnfsd	A filer daemon that permits PCs to mount filer file systems. The corresponding PC client software is called (PC) NFS.

PDC	Primary Domain Controller. The domain controller that has negotiated to be (or has been assigned as) the primary authentication server for the domain.
POST	Power-on self tests. The tests run by the filer after the power is turned on.
PVC	Permanent Virtual Circuit. A link with a static route defined in advance, usually by manual setup.
quota tree	A directory on which you can impose tree quotas, created by the `quota qtree` command.
RAID	Redundant Array of Independent Disks. A technique that protects against disk failure by computing parity information based on the contents of all the disks in the array. The filer uses RAID Level 4, which stores all parity information on a single disk.
RAID disk scrubbing	The process in which the system reads each disk in the RAID group and tries to fix media errors by rewriting the data to another disk area.
SCSI adapter	An expansion card that supports the SCSI disk drives and tape drives.
SCSI ID	The number of a disk drive on the SCSI chain (0-6).
SCSI address	The full address of a disk, consisting of the disk's SCSI adapter number and the disk's SCSI ID (for example, 9a.1).
serial adapter	An expansion card for attaching a terminal as the console on some filer models.

serial console	An ASCII or ANSI terminal attached to a filer's serial port. Used to monitor and manage filer operations.
share	A directory or directory structure on the filer that has been made available to network users and can be mapped to a drive letter on a CIFS client.
snapshot	An on-line, read-only copy of the entire file system that protects against accidental deletions or modifications of files without duplicating file contents. Snapshots allow users to restore files and allow the system administrator to back up the filer to tape while the filer is in use.
SVC	Switched Virtual Circuit. A connection established through signaling. The user defines the endpoints when the call is initiated.
system board	A printed circuit board that contains the filer's CPU, expansion bus slots, and system memory.
tree quota	A type of disk quota that restricts the disk usage of a directory created by the `quota qtree` command. Different from user and group quotas that restrict disk usage by files with a given UID or GID.
UID	User identification number.
VCI	Virtual Channel Identifier. A unique numerical tag defined by a 16-bit field in the ATM cell header that identifies a virtual channel over which the cell is to travel.
VGA adapter	Expansion card for attaching a VGA terminal as the console.

VPI	Virtual Path Identifier. An eight-bit field in the ATM cell header that indicates the virtual path over which the cell should be routed.
WAFL	Write Anywhere File Layout. The file system designed for the Network Appliance filer to optimize write performance.
WINS	Windows Internet Name Service.
workgroup	A collection of computers running Windows NT or Windows for Workgroups that is grouped for browsing and sharing.

Index

Numerics

8.3 format for file names 115

A

access control list (ACL)
    assigning rights to users 158
    being checked before UNIX permissions 151
    removing an entry from 158
access rights, assigning for CIFS shares 158
activity on the hot spare disk /etc/messages file 64
adding a foreign volume 77
adding disks 61
adding disks to a volume 76
administration host
    accessibility problems from 289
    modifying configuration files 13
    requirements for 6, 7
arp command 292
ATM
    network mask for 18
    PVCs
        incoming on the filer 101
        outgoing, establishing on the filer 100
        tearing down 104
AuthName directive 173
automatic e-mail to Network Appliance
    configuring 45–47
    contents of the messages 44
    designating the contact person 21
    using administration host for 6
autosupport daemon
    configuring with the options command 45–47
    how it works 43
autosupport options 31–32
autosupport.doit option 31, 47
autosupport.enable option 31, 45
autosupport.from option 31, 46
autosupport.mailhost option 31, 46
autosupport.noteto option 31
autosupport.to option 32, 46

B

backing up the filer 233–243
benefits of multiple volumes 68
block size for the restore command 248
blocking factor for dump command 234
boot types 279
booting
    and NVRAM 11
    /etc/rc script 17–20
    from floppy disk 278–291
    halt and reboot commands 11
    without /etc/rc file 279
broadcast filer address, setting using ifconfig 97

C

C$ share 8
cables, checking 292
CGI requests, redirecting 180
changing size of RAID groups 71
CIFS
  access rights, files 159
  adding clients 138
  assigning share-level access rights 158
  backups 231
  C$ share 8
  choosing file access rights 151–154
  creating shares 143
  creating users' home directories 142
  deleting a share 146
  displaying session information 164
  displaying share information 145
  displaying statistics 163
  effect of filer reboot on clients 11
  file access rights, choosing 151–154
  file ownership, changing 157
  file-naming conversions and NFS 117
  generic account 141
  guest access 140, 141
  guest account 140
  home directories for users, creating 142
  home directory share offer 146–147
  open files, limits 150
  oplocks 160–161
  options 32–34
  removing an ACL entry 158
  restoring files 245
  rsh use to enter filer commands 139
  session information, displaying 164
  sessions, limits 150
  shares
    changing description 144
    changing UNIX-style file permissions 154–155
    creating 143
    deleting 146
    effects of volume name change 149
    limits 150
    managing 143–146
  statistics, displaying 163
  stopping and restarting service 165
  symbolic links 108–111
cifs access command 158
cifs chmod command 154
cifs chown command 157
cifs mkdir command 142
cifs restart command 166
cifs rights command 159
cifs sessions command 164
cifs shares command 143
cifs stat command 163
cifs terminate command 165
cifs.generic_account option 32
cifs.guest_account option 32
cifs.home_dir option 33
cifs.oplocks.enable option 33
cifs.scopeid option 33
cifs.symlinks.cycleguard option 34, 111

cifs.symlinks.enable option 34, 109
clients
accessing snapshots from 207–210
NFS statistics in the custom MIB 82
per-client NFS statistics 133–136
collisions 292
commands
aborted by Ctrl-C 10
accessible through rsh 9
configuration
of volumes 67
planning for multiple volumes 69
configuration files
default 13–20
editing 7
errors in 290
/etc/dgateways 92, 94
/etc/exports file 120–122
/etc/hosts 85–87
/etc/hosts.equiv file 9
/etc/netgroup file 122
/etc/rc 17–20
/etc/resolv.conf 88
/etc/syslog.conf 25–27
configuration problems
booting with floppy disk for 290
filer accessibility 289
lost passwords 291
with /etc/rc 289–291
console session 9
copy-on-write technique 187, 195
core files 24
cpio, copying files with 296
crash files 19, 24
creating volumes 75
cron and crontab 48–49

D

Data ONTAP
initial volume configuration 66
Data ONTAP version, displaying 265
data rebuild, on the hot spare disk 286
data reconstruction
speed 65
when filer is shut down (degraded mode) 65
date command 48
default
configuration 13–16
route in routing table 92
router 18, 92
degraded mode 64
meaning of 73
reasons for 64
timeout period for automatic shutdown 64
when a hot spare disk is available 64
when a hot spare disk is not available 64
deprecated MIB objects 82
destroying a volume 77
df command 196–199, 229–230, 297
diagnostic messages 25–27, 277
directories
created by snapshots 207–208
exporting 19, 120
Directory directive 172
disk does not exist 284
disk fail command 73
disk in use message 284
disk information, displaying 266, 268

disk remove command 73
disk swap command 73
disk_erase_label command 72
disks
  adding 61, 62
  addressing, use of 58
  changing the size of RAID groups 71
  concepts 56
  degraded mode 64, 65, 73
  different sizes in same filer 61
  different types in RAID group 56
  failures, effects of 63, 65
  failures, handling 63
  free space, accessing information through SNMP 83
  free space, displaying 229–230
  freeing space by deleting snapshots 204
  hot swapping 60, 74
  information in the custom MIB 83
  installing new 71
  labels
    erasing 72
  management tasks 71
  maximum number of files 228
  overview 55
  parity 56, 62
  parity disk 229
  problems 286–287
  quotas 50, 83, 211, 213–226
  removing 73
  removing failed 73
  replacing 61, 62
  restricting usage 213–227
  right-sizing 62
  SCSI ID number 58
  setting size of RAID group 71
  snapshots, space used 195–206
  swapping 73
  usable space 62
DNS
  disabling 89
  domain name
    specifying with option 35
  domain name in /etc/rc 19
  enabled in /etc/rc 19
  enabling 88
  options 35
  querying the name server 87
  resolving names with 84
dns.domainname option 35
dns.enable option 35
double disk failures 286
dump
  blocking factor 241
  /etc/dumpdates 242
  /etc/dumpdates file 240, 243
  levels 240
  multiple subtrees 238, 243
  number of tapes requires 235
  size of file 234
  subtrees 239
  using filer tape drive from another system
    SVR4 UNIX 257
    UNIX (not SVR4) 258
  using snapshot 187
  volume size, maximum 241
dump command 233–243

E

editing configuration files 7
effects of disk failures 63

email support 277
Enabling 87
erasing disk labels 72
error handling
during hot swap 74
errors
caused by exceeding disk quotas 226
displayed by netstat 292
/etc/crash directory 19, 24
/etc/dgateways file 92, 94
/etc/dumpdates file 240, 242, 243
/etc/exports
exporting directories in 19
/etc/exports file
contents of 8, 120–122
deleting entries 122
exporting directories in 120–122
/etc/fstab file
balancing traffic among interfaces 271
editing 271
/etc/hosts file
accessing problems in 289
contents of 85
for IP address 18
/etc/hosts.equiv 9
/etc/httpd.access file 172
/etc/httpd.group file 173
/etc/httpd.hostprefixes file 176
/etc/httpd.passwd file 173
/etc/messages file
activity on the hot spare disk 64
configuring messages written to the file 25–27
files saved 27
for disk failures 286
messages from savecore 24
reporting to Network Appliance 277
warnings for disk failures 64
/etc/netgroup file 122–124
/etc/nsswitch.conf 84
/etc/nsswitch.conf file 14, 84
/etc/passwd file, copying for (PC)NFS 128
/etc/quotas file, configuring 214–218
/etc/rc file
commands in 17–19
damaged or missing 289–291
updating 17, 20
/etc/resolv.conf file 88
/etc/serialnum file 15
/etc/shadow file 15
/etc/shadow file, when using (PC)NFS 128
/etc/symlink.translations file 110
/etc/syslog.conf file 25–27
/etc/zoneinfo directory 19
Ethernet
interface names and numbers 22
network mask for 18
setting the media type 97
explicit routes in routing table 92
exportfs command 19, 122
exporting directories 120–122

F

FAX support 277
FDDI
interface names and numbers 22
network mask for 18
setting the MTU 98
file locking, differences between NFS and CIFS 107

file names
conventions 117
conversion 114
legal characters 115
maximum length 114
file names used by NFS and CIFS clients 115
file space, incorrect display 297
file system
damaged 245, 291
exporting 120–122
inconsistent 287
installing a new file system 291
maximum number of files 228
mounting 294
protection through RAID scrubbing 58
filer information, overall, displaying 267
filer password
changing 10
filer system load
systat command 65
filer, restarting 280
files
access rights, CIFS 159
copying with cpio 296
ownership
changeable by root only 50
changing, CIFS 157
same file criteria 188
setting maximum number of 228
firewall, virtual 175

G

generic account, CIFS 141
group quotas, creating 223
guest
access, CIFS 140
account, CIFS 140

H

halt command 11, 281
home directories for CIFS users 142
/home directory
exporting 121
read and write permissions 121
host names
for network interfaces 22
resolving 85–90
hostname command 18
hot spare
add or remove when not supported 74
hot spare disk 72
availability
sysconfig command 65
overview 60
removing 73
replacement activity (/etc/messages) 65
hot swap error handling 74
hot swapping a disk 60, 74
hot swapping, overview of 60
hourly snapshots 190, 192
HTTP
displaying connection information 184
displaying statistics 185
options 35–36
password protection 172–173
root directory 171
starting service 170
virtual hosting, enabling 176
httpd.admin.enable option 35
httpd.enable option 35

httpd.log.max_file_size option 35
httpd.rootdir option 36
httpd.timeout option 36
httpd.timewait.enable option 36

I

ICMP redirect messages 92
identifying disks
    SCSI ID 58
IERRS (input errors) displayed by netstat 292
ifconfig command 17, 18, 86, 97–99, 292
illegal volume name message 283
inconsistent file system 287
inodes, effects of maximum number of files on 228
installing new disks 71
interfaces
    errors on 292
    how packets are sent and received 95
    using ifconfig to configure 97–99
    See also ATM, Ethernet, FDDI
invalid volume name message 283
IP addresses
    setting in /etc/rc file 18
    setting using ifconfig 97
ip.path_mtu_discovery.enable option 41

L

legal characters in file names 115
license command 51, 53
licenses 51–53
limitations of multiple volumes 68
logging in to the filer 9
logging out of the filer 9
lost data from disk failures 286
lost passwords 291
ls command, listing snapshot files 208, 210

M

mail host
    automatic email 6, 43
    designating other mail hosts 46
Makefile, NIS 86, 123
making a volume inactive 76
management tasks
    for disks 71
    for volumes 75
managing disks, overview 55
maxfiles command 228
maximum number of files 228
media type for an Ethernet interface 97
messages
    configuring the syslogd daemon 25–27
    file of 25–27
    for disk failures 286
    panic 298
    severity of 25
MIB objects
    multivolume, locations 82
MIB objects, deprecated 82
MIB, Network Appliance Custom 82
MIB-II 80
MIME Content-Type, specifying 178
minra option 272
minra volume option 39
monitoring status
    of volumes 76
mount_rootonly option 50

mounting files
if there are quota trees 230
problems with 294
restricting mount privilege to root 50
mounting volumes 67
mt command 262
multiple RAID groups 56
multiple volumes
benefits of 68
configuration planning 69
limitations of 68
multivolume MIB objects, locations 82

N

name services, specifying the order in which contacted 84
names
resolving 84–91
volume naming conventions 66
NDMP 259–261
ndmp command 260
NetBIOS scope IDs 162
netstat command 92, 95, 273, 292
network interfaces
balancing traffic among 271
configuring 97–99
naming conventions 22
network statistics, displaying 273
networks
connections, checking 292
displaying statistics (netstat command) 273
how filer sends and receives traffic 95
management services, using SNMP 80–83
network mask
configuring using ifconfig 97
in /etc/rc file 18
problems with 292
statistics 292
using ifconfig to configure 97
NFS
displaying per-client statistics using the nfsstat command 133
file-naming conversions and CIFS 117
how interfaces respond to packets 95
problems with 294
statistics in custom MIB 82
statistics, displaying (nfsstat command) 132
turning on 20
NFS and (PC)NFS options 36–38
nfs command 20
NFS over UDP requests 95
nfs.big_endianize_fileid option 272
nfs.mount_rootonly option 36
nfs.per_client_stats.enable option 36, 133
nfs.tcp.enable option 37
nfs.v2.df.2gb.lim option 37
nfs.v2.df_2gb_lim option 36, 37, 297
nfs.v3.enable option 37
nfs.webnfs.enable option 37, 130
nfs.webnfs.rootdir option 37, 131
nfs.webnfs.rootdirset option 38, 131
nfsstat command 132–135
nightly snapshots 190–192
NIS
changing domain name 91

disabling 90
domain name
set in /etc/rc 19
specifying with option 38
enabled in /etc/rc 19
enabling
during setup 90
without using setup 90
maps supported 89
options 38
propagating changes
/etc/hosts on filer 86
/etc/netgroup on filer 123
/etc/passwd on filer 128
/etc/shadow for (PC)NFS on filer 128
nis.domainname option 38
nis.enable option 38
no_atime_update option 272
no_atime_update volume option 40
nonexistent disks 284
Nonvolatile RAM (NVRAM)
batteries 64, 281
cache 3
failures in 277
inconsistent contents 281
logging requests 11
when to increase 269
nosnap option 211
nosnap volume option 40
nosnapdir option 211
nosnapdir volume option 40
numbering network interfaces 22

O

oplocks 160–161
options
autosupport.doit 31, 47
autosupport.enable 31, 45
autosupport.from 31, 46
autosupport.mailhost 31, 46
autosupport.noteto 31
autosupport.to 32, 46
cifs.generic_account 32
cifs.guest_account 32
cifs.home_dir 33
cifs.oplocks.enable 33, 161
cifs.scope.id 162
cifs.scopeid 33
cifs.symlinks.cycleguard 34, 111
cifs.symlinks.enable 34, 109
dns.domainname 35
dns.enable 35
httpd.admin.enable 35
httpd.enable 35, 170
httpd.log.max_file_size 35, 170
httpd.rootdir 36, 170, 176
httpd.timeout 36
httpd.timewait.enable 36
ip.path_mtu_discovery 41
minra 272
mount_rootonly 50
nfs.big_endianize_fileid 272
nfs.mount_rootonly 36
nfs.per_client_stats.enable 36, 133
nfs.tcp.enable 37
nfs.v2.df_2gb_lim 36, 37, 297
nfs.v3.enable 37
nfs.webnfs.enable 37, 130
nfs.webnfs.rootdir 37, 131
nfs.webnfs.rootdirset 38, 131
nis.domainname 38
nis.enable 38
no_atime_update 272
nosnap 211

nosnapdir 211
pcnfsd.enable 38, 127
pcnfsd.umask 38
raid.reconstruct_speed 39, 65
raid.scrub.enable 39, 59
raidtimeout 39, 63, 64
root_only_chown 50
rps.status 41
telnet.hosts 41, 50
volume options
minra 39
no_atime_update 40
nosnap 40
nosnapdir 40
raidsize 40
root 40
wafl.maxdirsize 41, 271
wafl.root_only_chown 42
options command
making changes permanent 28
See also options
using 28

P

packets, responses to 95
panic
files 24
message 298
parity disk 56, 62
parity disks
role 57
passwd command 10
password
HTTP pages 172–173
passwords
changing with passwd command 10
lost 291
used for shell session 9
(PC)NFS 127–128
pcnfsd daemon 127–128
pcnfsd.enable option 38, 127
pcnfsd.umask option 38
performance, improving 269
permissions
for snapshots 188
on exported directories 122
ping command 292
protocol licenses 51–53
PVCs, ATM
establishing on a remote ATM host 102, 104
establishing on interconnecting ATM switches 102
incoming, establishing on the filer 101
outgoing, establishing on the filer 100

Q

quota
command 218
disk, setting up 213–226
displaying report 221
effects of snapshots on 211
effects on clients when exceeded 226
resizing 220
tree, creating 224
tree, restoring 246
quotas, disk
information available through SNMP 83
information available through the custom MIB 83

when changing file ownership 50

R

RAID
displaying information for each group 268
displaying overall information 266
RAID (Redundant Array of Independent Disks)
accessing information through SNMP 83
adding disks 61
data reconstruction speed 65
disk scrubbing 58, 59
displaying information about disks 266, 268
group size characteristics 57
groups 56, 66
information in the custom MIB 83
options 39
spare disk use 56
support for multiple groups 56
RAID information, overall, displaying 266, 268
raid.reconstruct_speed option 39, 65
raid.scrub.enable option 39, 59
raidsize volume option 40
raidtimeout command 64
raidtimeout option 39, 63
rdate command 48
reactivating an off-line volume 77
read-ahead, minimal 272
read-only bit 112–113
reboot command 11
rebooting
and NVRAM 11
automatic email to Network Appliance 24
downed filer 280
from floppy disk 278–288
halt and reboot commands 11
scripts for 17
to update configuration 20
remote shell (rsh)
accessing the filer from the administration host 9
for entering filer commands 9–10
removing
a failed disk 73
a hot spare disk 73
removing volumes from a filer 69
renamed volume not exported 283
renaming volumes 77
replacing
disks 62
failed disks 61
requests, replies to 95
require group directive 173
require user directive 173
resolving host names 85
restarting a filer 280
restore command 245–253
restore_symboltable file 251
restoring files, remote access to a filer's tape drive
SVR4 UNIX 257
UNIX (not SVR4) 258
restoring the entire filer 251
right-sizing disks 62
role of parity disks 57
root volume 7, 66
root volume option 40

root_only_chown option 50
route command 92
routed command 93
routed daemon
/etc/dgateways file 94
purpose of 93
router
default 18, 92
problems with 292
Routing Information Protocol (RIP) 93
routing table, filer 92
rps.status option 41
rsh. See remote shell (rsh)

S

savecore command 19, 24
SCSI ID, identifying disks 58
security
mounting /home 122
options for 50
permissions on /home 121
system password 10
setting up the filer 13
shadow file 15
shares, managing 143–146
shutdown filer, restarting 280
single disk failures 286
size of dump file 234
snap command 189–193
snapshot_for_dump snapshot 239
snapshots
accessing 207–208
automatic 19, 190–192
commands for 189
definition 187
deleting to free space 204
directory name displayed on CIFS clients 209
disk consumption by 196–206
effects on disk quotas 211
information in the custom MIB 83
ls command 208, 210
magic directories 208
making snapshot directory invisible 211
operation of 194–195
options 39–40
reserving space for 197–199, 201–203, 204
scheduling 200
snapshot_for_dump file 239
temporarily disabling 211
types 190
SNMP
commands
examples of 80–81
in /etc/rc 20
configuring the agent 80
custom MIB 82
snmp command 80–81
software licenses 51–53
spare disks in RAID groups 56
sticky bit 15
subnets, exporting to 124–125
svtx bit 15
swapping out disks 73
symbolic links, CIFS 108–111
sysconfig command 266–267
hot spare disk availability 65
usable disk space 61
sysconfig -d command 266
sysconfig -r command 266
sysconfig -t command 266
sysconfig -v command 277

syslogd daemon 25
sysstat command 269–270
  for filer system load 65
system boot diskette 278
system crashes 19, 24
system message files 25–27
system panics 24, 298

T

tape drive information, displaying 266
tape drives
  controlling 262–264
  displaying information about 266
  remote access to filer 257
tapes, number required by dump 235
TCP/IP, how interfaces respond to packets 96
technical assistance, getting 277
Technical Support
  contacting 277
  email address 43
telnet connection to the filer
  administration host 7
  limiting host access 41, 50
telnet.hosts option 41, 50
time
  setting on the filer 48
  synchronizing the filer with another system 48
time zone, setting 19
timezone command 19
troubleshooting
  configuration problems 289–291
  cpio problems 296
  df problems 297
  disk problems 286–287
  downed filer 280
  help for 298
  network problems 292
  NFS problems 294
  NVRAM problems 281–282
  UNIX cpio problems 296
  UNIX df problems 297
  volume problems 283–285
trusted host 10

U

unrecognized volume name message 283
uptime command 270
URL, how filer translates 180
usable space on disks 62
use of hot spare disks 60
user quotas, creating 222

V

valid volume names 284
version command 265
version of Data ONTAP, displaying 265
virtual firewall 175
virtual host addresses, mapping 177
virtual hosting, enabling 176
vol command
  add 76
  create 71, 75
  destroy 77
  offline 76
  online 77
  options 76
  options raidsize 71
  rename 77
  status 76

vol options command
using 28
vol status command 268
vol status -d command 268
vol status -r command 268
vol status -v command 268
volume information, displaying 268
volume name not valid message 283
volume names 7
volume not exported 283
volume options
minra 39
nosnap 40
nosnapdir 40
raidsize 40
root 40
volume state information, displaying 268
volumes
adding disks 76
adding foreign volume 77
concepts 66
configuration of 67
creating of 75
destroying 77
handling failures 70
making inactive 76
management tasks 75
monitoring status 76
mounting 67
naming conventions 66
problems with 283–285
reactivating 77
removing 69
renaming 77
setting options 76

W

wafl.maxdirsize option 41, 271
wafl.root_only_chown option 42
warnings for disk failures 64
WebNFS 130–131
weekly snapshots 190–195

Y

ypwhich comand 91