Dell™ PowerVault™ 720N, 740N, and 760N

SYSTEM ADMINISTRATOR AND COMMAND REFERENCE GUIDE

Information in this document is subject to change without notice.
© 1998–1999 Network Appliance, Inc. Licensed to Dell Computer Corporation. All rights reserved.

Reproduction in any manner whatsoever without the written permission of Dell Computer Corporation is strictly forbidden.

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 used in this text: *Dell*, the *DELL* logo, and *PowerVault* are trademarks of Dell Computer Corporation; *Data ONTAP, SnapMirror, SnapRestore, Snapshot, WAFL, FilerView,* and *SecureShare* are trademarks of Network Appliance, Inc.*; MS-DOS, Microsoft, Windows*, and *Windows NT* are registered trademarks and *Windows for Workgroups* is a trademark of Microsoft Corporation; *UNIX* is a registered trademark of The Open Group in the United States and other countries.

Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Dell Computer Corporation disclaims any proprietary interest in trademarks and trade names other than its own.

April 1999 P/N 4385P Rev. A00

Preface

About This Guide

This guide describes how to configure, operate, and manage Dell PowerVault F720N, F740N, and F760N filers that run Data ONTAP 5.3 software. The guide is organized in three parts:

- Chapters 1 through 19 describe how to configure, operate, and manage Dell PowerVault 720N, 740N, and 760N filers.
- Appendix A, "Command Reference," provides the commands that you use to control a filer.
- "Glossary" provides definitions of terms, acronyms, and abbreviations used in this guide.

Audience

This guide is for system administrators who are familiar with operating systems that run on the filer's clients, such as UNIX, Microsoft Windows 9*x*, and Microsoft Windows NT. It also assumes that you are familiar with how the Network File System (NFS), Common Internet File System (CIFS), and Hypertext Transfer Protocol (HTTP) protocols are used for file sharing or transfers. This guide doesn't cover basic system or network administration topics, such as Internet Protocol (IP) addressing, routing, and network topology; it emphasizes the characteristics of the Dell filer.

Other Documents You May Need

Besides this *System Administrator and Command Reference Guide*, the following documentation is included with your system:

- The *Getting Started* document provides step-by-step instructions for setting up your computer system.
- The *Quick Reference* card provides the filer commands and command options.
- The *Installation and Troubleshooting Guide* provides instructions for installing system hardware and includes troubleshooting and diagnostic procedures for testing your computer system.

- The *User's Guide* provides instructions for configuring and operating a new filer that runs Data ONTAP 5.3 software.

You may also have one or more of the following documents.

NOTE: Documentation updates are sometimes included with your system to describe changes to your system or software. Always read these updates ***before*** *consulting any other documentation because the updates often contain the latest information.*

- Documentation is included with any options you purchase separately from your system. This documentation includes information that you need to configure and install these options. Installation instructions for the options are included in the *Installation and Troubleshooting Guide*.
- Technical information files—sometimes called "readme" files—may be installed on your root volume to provide last-minute updates about technical changes to your system or advanced technical reference material intended for experienced users or technicians.

Terminology

This guide uses the following terms:

- *Filer* refers to a PowerVault F720N, F740N, or F760N storage system.
- *System* refers, at a minimum, to a filer and a connected PowerVault F700N Disk-Array Enclosure (DAE) storage system. A tape backup device can also be a component of the system, but is not required.

Notational Conventions

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.

Key Combinations

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.

Typographical Conventions

Typographical conventions used in this guide are shown in the following table:

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 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.

Contents

Figures

Tables

/

CHAPTER 1
Introducing Dell™ Filers

About Filers

What a Filer Is

A filer is a hardware and software system. It acts on network requests from clients and processes them by writing data to or retrieving data from disks in PowerVault 700N Disk-Array Enclosure (DAE) storage systems that are connected to it. The software that enables the filer to perform these tasks is the Data ONTAP™ 5.3 operating system.

Components of a Filer

A filer consists of the following major components listed in Table 1-1.

Table 1-1. Major Components of a Filer

Component	Function
Filer main unit	The piece of hardware that receives and sends data.
PowerVault 700N storage systems	Hardware that holds disks and is connected to the filer.
Data ONTAP 5.3	Software that is the operating system for the filer.

Filer and Filer Main Unit

Often, the filer main unit is referred to simply as a filer.

What a Filer Does

A filer provides the features described in Table 1-2.

Table 1-2. Filer Features

Feature	Description
Network file service	The filer enables users on client workstations to create, delete, modify, and access files stored on it. Client workstations are connected to the filer through network connections.
Multiprotocol file sharing	Clients can use the following protocols to access data on the filer: • CIFS (Common Internet File System)—used by Windows clients. • HTTP (HyperText Transmission Protocol)—used by the World Wide Web. • NFS (Network File System)—used by UNIX® systems. Files written using one protocol are accessible to clients of any protocol, provided that filer licenses and permissions allow it. For example, an NFS client can access a file created by a CIFS client, and a CIFS client can access a file created by an NFS client.
Data protection	The filer protects disk data in the following ways: • Network Transaction Logging—the filer records network transactions in case of failures and reconstructs transactions on recovery. • Disk Redundancy—the filer reconstructs data disks in case of disk failure.
Autosupport	Mail notifications about filer problems is automatically sent to the customer-defined administrator accounts.

How You Administer a Filer

You administer the filer using the Data ONTAP 5.3 operating system. You can use the following methods of administering a filer:

- Command execution through the filer's command line
- Command execution through Microsoft® Windows NT® operating system
- Configuration file editing
- Command execution through FilerView

Command Execution Through the Filer's Command Line

You use the filer's command line to execute all Data ONTAP 5.3 administrative commands from the command line, with the exception of some Windows NT administrative commands.

You can access the filer's command line from

- A serial terminal connected to the Console port of the filer
- A `telnet` session to the filer

You can also access some commands through a remote shell program, such as the UNIX `rsh` utility.

Command Execution Through Windows NT

You use Windows NT commands to perform filer administrative tasks related to Windows NT.

You can execute Windows NT commands that affect the filer using native Windows NT administration tools such as Server Manager and User Manager.

Configuration File Editing

You edit configuration files to supply information that Data ONTAP 5.3 needs to perform certain tasks.

You can access configuration files by mounting the root directory of the filer on a UNIX client or by mapping the administrative share (C$) to a drive on a Windows client, then editing the file from the client.

Command Execution Through FilerView

You use FilerView to perform most administrative tasks from a Web-based interface.

You can use FilerView even if you did not purchase a license for the HTTP protocol.

About Filer Main Unit Components

Two Kinds of Components

The filer main unit has two kinds of components: internal components that enable it to function, and slots and ports that connect it to networks and PowerVault 700N storage systems.

Internal Filer Components

The internal components described in Table 1-3 enable the filer to function.

Table 1-3. Filer Internal Components

Component	Description
System board	The system board is the main board of the filer. It has upgradable firmware. All components are connected to the system board.
System memory	System memory stores information temporarily.
NVRAM (Nonvolatile Random Access Memory)	Data ONTAP 5.3 uses NVRAM to log network transactions as a data integrity measure. In case of a system or power failure, Data ONTAP 5.3 uses the contents of NVRAM to restore network data to disk.
Diskette drive	In an emergency and for major upgrades, Data ONTAP 5.3 uses the diskette drive to boot from diskettes.
LCD and LEDs	The filer displays status information on the LCD and LEDs.
Environmental adapter	The environmental adapter performs the following functions: • Monitors the filer's temperature and fans • Sends critical information to the filer's LCD • Logs information • Shuts down the filer if its temperature is beyond a critical range or the fans cease operating

Slots and Ports

The filer has slots for external connections and ports for a console and diagnostic hardware, as shown in Table 1-4.

Table 1-4. Slots and Ports

Component	Description
Slots	The filer contains expansion slots for network interface cards, tape drive adapters, and PowerVault 700N storage system adapters.
Serial ports	The two serial ports are as follows: • The console port connects to the filer a serial terminal that you can use as a console. • The diagnostics port is not used.

About PowerVault 700N Storage Systems

PowerVault 700N Storage Systems Contain Disks

A PowerVault 700N storage system contains the disks that store the data that the filer serves.

PowerVault 700N Storage System Environmental Information

PowerVault 700N storage systems collect information about the presence of disks, fan status, power supply status, and temperature. PowerVault 700N storage systems send messages to the console if parameters exceed permissible operating conditions.

About Data ONTAP 5.3

Data ONTAP 5.3 Overview

You administer the filer using Data ONTAP 5.3 commands.

Data ONTAP 5.3 manages data in the following three ways:

- Data storage management
- Data organization management
- Data access management

Data Storage Management

Data ONTAP 5.3 stores data on disks in PowerVault 700N storage systems. Disks are organized into RAID groups, and RAID groups are organized into volumes. These items are explained in Table 1-5, along with what aspects of them you can administer.

Table 1-5. Data Storage Management Concepts

Data Storage Concept	Explanation	Storage Administrative Actions
RAID (Redundant Array of Independent Disks)	A feature that enables file access even if one disk in a RAID group is damaged.	
RAID group	A RAID group consists of a parity disk and up to 27 data disks, and optional spare disks.	You can control the size of a RAID group. This enables you to customize backups and disk failure recovery.
Data disks	Hold the data that clients access.	You can control the number of data and spare disks that a filer can use. This enables you to manage recovery if a disk fails.
Parity disk	Contains information that Data ONTAP 5.3 uses to reconstruct data if a data disk fails. Each RAID group has one parity disk.	
Spare disks	Replace failed data disks automatically with reconstructed data.	
Volumes	Pools of storage composed of multiple disks that store client data. A filer can support up to 23 volumes or can have only one volume.	You can specify the size of a volume, add disks to it, copy volumes, and control the number of RAID groups in a volume. This gives flexibility in managing data storage tasks such as backups and restores, and enables you to customize volumes.

Data Organization Management

Data is organized in files, which are the smallest unit of data management. Users organize files into directories, and you organize directories into file systems, which are known as volumes. You can also organize directories into special directories called qtrees. Major data organization concepts that are special to filers are explained in Table 1-6.

Table 1-6. Data Organization Management Concepts

Data Organization concept	Explanation	Data Organization Administrative Actions
volume	An independent file system.	You can specify the following features for a volume: • A security style, which determines whether a volume can contain files that use UNIX security, Windows NT File System (NTFS) security, or both type of files • Whether it uses CIFS oplocks • Disk space and file limits This enables you to customize volumes for the needs of your users.
qtree	A special subdirectory of the root directory of a volume. It has the following special attributes: • A security style • A CIFS oplocks setting • Disk space and file limits	You can specify the following features for a qtree: • A security style like that of volumes • Whether it uses CIFS oplocks • Disk space and file limits This enables you customize areas for projects and to keep users and projects from monopolizing resources.

Data Access Management

Data ONTAP 5.3 manages access to data by performing the following operations:

- Checks file access permissions against file access requests.
- Checks write operations against file and disk usage quotas that you set.
- Takes snapshots and makes them available so that users can access deleted or overwritten files. Snapshots are read-only copies of the entire file system.

Data ONTAP 5.3 enables you to perform the following actions:

- Administer network connections
- Administer protocols
- Dump data to tape and restore it to the filer

- Copy volumes
- Mirror volumes

Filer Administration With Data ONTAP 5.3

Filer Administration Activities

Administering a filer involves the following activities:

- Configuring the filer
- Monitoring and maintaining client access to data
- Monitoring and maintaining network access
- Monitoring and maintaining filer hardware
- Performing periodic administration tasks

Configuring the Filer

You configure the filer by running either the Setup Wizard or the `setup` program or both if necessary, then editing configuration files if necessary.

Monitoring and Maintaining Client Access

You monitor client access to data on the filer by the following methods:

- Gathering network statistics so that you can verify and improve the performance of the filer
- Gathering file statistics so that you can schedule snapshots
- Monitoring accesses to a file by CIFS clients so that you can monitor potential security problems

You maintain client access by the following methods:

- Configuring volumes and qtrees to accommodate the needs of Microsoft® Windows operating systems and UNIX clients
- Configuring volumes so that there is enough disk space for data
- Configuring RAID groups to maintain maximum data availability
- Configuring snapshots so that data is recoverable in case of accidental deletions
- Setting quotas to make sure that users have enough resources for their work

Monitoring and Maintaining Network Access

You monitor network access by gathering network statistics so that you can verify and improve the performance of the filer and its network interfaces.

You maintain network access by following general filer network administration procedures and procedures specific to your protocol. These procedures are described in later chapters in this guide.

Monitoring and Maintaining Filer Hardware

You monitor hardware by gathering statistics about your hardware and analyzing them for performance.

You maintain the hardware as described in the appropriate hardware guides.

Periodic Administration Tasks

Table 1-7 lists essential tasks that you should consider performing periodically, suggests how often to perform a task, and lists the command or file to use for the task.

Table 1-7. Periodic Administration Tasks

How Often	Task	Command or File
Daily	Review filer performance and CPU utilization.	`sysstat` command
	Review filer disk usage on a per-volume and per-snapshot level.	`df` command
	Check for drive failures, reboots, and other logged events	*/etc/messages* file
	Review CIFS utilization, if licensed for CIFS.	`cifs stat` command
	Review HTTP utilization, if licensed for HTTP.	`httpstat` command
	Review NFS utilization, if licensed for NFS.	`nfsstat` command

Table 1-7. Periodic Administration Tasks *(continued)*

How Often	Task	Command or File
Weekly	Review filer disk usage on a per-user level.	`quota report` command
	Test connectivity to NT Domain Controller.	`cifs testdc` command
	Review current shared CIFS directories.	`cifs shares` command
	Review NFS exports	*/etc/exports* file
	Review filer network traffic statistics.	`netstat` command
	Review network interface performance statistics on a per-interface level.	`ifstat` command
Monthly	Make sure that the PowerVault 700N storage systems are connected properly.	`shelfchk` command
	Review filer uptime.	`uptime` command
	Review filer OS revision.	`version` command
	Review current NIS server.	`ypwhich` command

CHAPTER 2
Filer Administration Basics

Overview

About This Chapter

This chapter describes routine filer administration procedures that you need 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.

Using the Administration Host

About the Administration Host

The filer recognizes a single client computer as the administration host. The administrator who set up the filer specified the name of the administration host using the `setup` program.

Administration Host Privileges

The filer granted root permissions to the administration host after the `setup` procedure was completed. Table 2-1 describes the administration host's privileges.

Table 2-1. Administration Host Privileges

If the administration host is...	You can...
an NFS client	• Mount the filer root directory and edit configuration files from the administration host. • Enter filer commands by using a remote shell program such as `rsh`.

Table 2-1. Administration Host Privileges *(continued)*

If the administration host is...	You can...
a CIFS client	Edit configuration files from any CIFS client as long as you connect to the filer as root or "Administrator."

Administration Host Entry in the /etc/hosts.equiv file

The `setup` procedure placed the administration host name in the */etc/hosts.equiv* file automatically.

Administration Host as the Mail Host

For the administration host to send email, it must provide a server for the SMTP protocol, such as the `sendmail` program, or the Microsoft Exchange server.

Designating a Different Mail Host

You can designate another host at your site to be the mail host at any time. Refer to "Use the Options Command to Configure Autosupport" for information about how to specify a different mail host.

Requirements for Using an NFS Client as the Administration Host

If you plan to use an NFS client to manage the filer, the client must

- support a text editor that can display and edit text files containing lines ending with the newline character
- support the `telnet` and `rsh` commands
- be able to mount directories using the NFS protocol

Requirements for Using a CIFS Client as the Administration Host

If you plan to use a CIFS client to manage the filer, the client must support the `telnet` and `rsh` commands.

The Root Volume

About the Root Volume

Every filer has a root volume. It is the volume from which the filer reads configuration files. During `setup`, the filer creates a default volume, named *vol0*, and designates it as the root volume.

Designating the Root Volume

If you add volumes to your filer, you choose a volume to be the root volume during the multivolume configuration process.

For more information about designating the multiple volume configuration process, read Chapter 3, "Disk and File System Management."

About the Volume Name Prefix

Volume names begin with the following prefix:

/vol/

Syntax to Refer to the Root Volume From NFS Clients

To refer to the root volume when mounting the root volume to an NFS client, use the following syntax:

/vol/vol0

Editing Configuration Files

What Editor to Use

The filer does not include a local editor. You must use an editor from a client to change filer configuration files.

Where Configuration Files Reside

Configuration files reside in the */etc* directory in the filer's root volume.

Choosing an NFS or a CIFS Client

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.

Editing Files From an NFS Client

Table 2-2 describes how to edit configuration files from an NFS client.

Table 2-2. Editing Configuration Files From a NIFS Client

Step	Action	
1	**If the NFS client is...**	**Then...**
	the administration host	Mount the filer root volume to the host.
	not the administration host	1. Mount the filer root volume to the administration host. 2. From the administration host, edit the */etc/exports* file on the root volume to grant root permission to the client. 3. Use the filer console, a `telnet` client, or the `rsh` command to issue the following command to the filer: `exportfs` 4. Mount the filer root volume to the client.
2	From the client, use a text editor to edit the files in the */etc* directory.	

Editing Files From a CIFS Client

To edit configuration files from a CIFS client, perform the following steps:

1. Connect from a CIFS client to the filer as Administrator.

   After `setup` finished, the default */etc/passwd* and */etc/group* files on the root volume were set up to enable you to share files on the filer as Administrator.

2. Display the contents of the filer's *C$* share and select a file to edit.

   After `setup` finished, the filer root directory was shared automatically as *C$*. The Administrator 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

Ways to Access the Command Line

The filer supports a command-line interface. You can access the command line:

- directly, from the system console
- remotely, using `telnet`
- remotely, using a remote shell such as `rsh`

Sharing a Single telnet and Console Session

The console and `telnet` share a single session. Everything entered through `telnet` is echoed at the console; everything entered at the console is echoed to the `telnet` session.

telnet Session Restriction

Only one `telnet` session can be open at a time.

Closing a telnet Session

To close a `telnet` session, press Ctrl-] to log out of the filer, then press Ctrl-D to log out of `telnet`.

telnet and Console Password Requirement

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

rsh Support

The filer supports `rsh` with trusted remote hosts—those remote hosts listed in */etc/hosts.equiv* on the root volume.

In addition to entering `rsh` commands manually, you can use a shell script or `crontab` file to enter some commands.

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

Commands Accepted From rsh

Table 2-3 lists the filer commands that you can execute

Table 2-3. Commands Accepted From rsh

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

Use Ctrl-C to Terminate the Command That Is Running

Ctrl-C terminates whatever command is being run from the console or a `telnet` session.

Because the console and `telnet` share a single session, a command entered at the console or through `telnet` can be terminated inadvertently from either location.

To ensure that a command is not terminated by Ctrl-C, start the command from a trusted host through `rsh`.

Changing the System Password

A system password is required to establish a console or `telnet` connection with the filer. The password was specified during `setup`.

You can change the system password at any time with the `passwd` command. When you enter the `passwd` command, the filer prompts you to enter the old password, if any, and then requests the new password twice.

Where to Go to Learn More About Security

For information about ways to increase filer security in addition to password protection, use the options described in "Using Options Command Options to Maintain Filer Security."

About Multiple Administrative Users

What Is an Administrative User?

An administrative user is a named account that exists on a filer. Administrative users have the same privileges as root, but can have a different password than root.

Multiple Administrative Users Increase Filer Security

Adding multiple administrative users to a filer means you no longer need to share the root password. Instead, administrative users access the filer, either locally or remotely, with a unique login name and password. The filer records each user name at login with a syslog message in */etc/messages*, to enable auditing.

Command to Use to Create Administrative Users

Use the `useradd` option of the `useradmin` command to create multiple administrative users on a filer.

Ways to Access the Filer Using an Administrative Login Name

Using an administrative login name, you can access a filer through any of the following tools:

- Filer console
- `telnet`
- `rsh`

Creating Administrative Users

Description

The `useradmin` command enables you to increase filer security by creating multiple administrative users, rather than sharing the root name and password among multiple administrators.

Prerequisites

You must be logged in as root or as an existing administrative user to use the command.

Restrictions

The user name is sensitive. It cannot contain any of the 15 characters shown in Table 2-4.

The password should contain at least six characters, including at least two alpha characters and one numeric or special character.

Table 2-4. Character Restrictions for User Name

Character	Character
* (asterisk)	< (less than sign)
\ (back slash)	| (pipe)
: (colon)	+ (plus sign)
, (comma)	? (question mark)
= (equal sign)	] (right bracket)
/ (forward slash)	; (semicolon)
> (greater than sign)	space
[(left bracket)	

Steps to Create a New Administrative User Using a Console or Telnet

To create a new administrative user using the filer console or a `telnet` session, perform the following steps:

1. Enter the following command:

   `useradmin useradd` ***`user_name`***

   `user_name` is the new administrative user login name.

2. Enter a password for the new user when prompted.

Step to Create a New Administrative User Using rsh

To create a new administrative user using `rsh`, enter the following command:

`useradmin useradd` ***`user_name password`***

user_name is the new administrative user login name.

password is the new password associated with the new user name.

Deleting Administrative Users

Description

Use the `userdel` option to delete administrative users created with the `useradd` option of the `useradmin` command. You can use the `userdel` option at any time to maintain filer security by keeping the administrative user list current.

Caution

You are not prompted for a password when deleting a user.

Step

To delete an administrative user, enter the following command:

`useradmin userdel` ***`user_name`***

user_name is the login name you want to delete.

Listing Administrative Users

Description

Use the `userlist` option to get a list of administrative users on a filer. You can use the `userlist` option at any time to determine whether you need to add or delete administrative users.

Step

To list one or more administrative users using a console, `telnet` or `rsh`, enter the following command:

`useradmin userlist` ***`[username_list]`***

username_list is a space-separated list of login names you want to display.

NOTE: If you do not specify any user names, all administrative users are listed.

Changing an Administrative User Password

Description

Use the `passwd` command from a console, `telnet`, or `rsh` to change your administrative password on a filer at any time. Changing your password at regular intervals can provide you with better filer security.

Restrictions

When you use `telnet` or a console, the `passwd` command prompts you for the user name if administrative users, other than root, exist.

The `passwd` command only accepts arguments when you use `rsh`.

Steps to Change an Administrative Password Using a Console or telnet

To change an administrative user password using a `console` or `telnet`, perform the following steps:

1. Enter the `passwd` command.
2. When prompted, enter the administrative user name of the user whose password you want to change.
3. Enter the old password.
4. Enter the new password.

Step to Change an Administrative Password Using rsh

To change an administrative user password using `rsh`, enter the following command:

`passwd` *`old_password new_password username`*`.`

NOTE: If you do not use the third argument, the `passwd` *command works as before, and expects you to enter the old and new password for root.*

Halting and Rebooting the Filer

Data Storage in NVRAM

The filer stores requests it receives in nonvolatile random-access memory (NVRAM). The use of NVRAM

- improves system performance
- prevents loss of data in case of a system or power failure

NVRAM Event During Orderly Shutdown

The `halt` and `reboot` commands perform an orderly shutdown. During an orderly shutdown, the contents of NVRAM are flushed to disk.

Procedure to Halt the Filer

To halt the filer, enter the following command:

```
halt
```

The filer displays the following prompt:

```
ok
```

Procedure to Boot the Filer

To boot the filer, perform the following steps:

1. Ensure that the **ok** prompt is displayed on the console.
2. Enter the following command:

   boot

Procedure to Reboot the Filer

You can halt and reboot the filer in a single operation by entering the following command:

```
reboot
```

Where the Filer Boots From

When the filer boots, it uses the boot diskette in its diskette drive, if there is one. Otherwise, the filer boots from its hard disk.

Use the Halt Command to Avoid Data Loss

You should always execute the `halt` command before turning the filer Off for the following reasons:

- 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.
- 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.

CAUTION: Never interrupt CIFS service by halting the filer without giving advance warning to CIFS users.

Before turning the filer Off, use the `halt` or `cifs terminate` command to send a warning message to CIFS users. This method gives users an opportunity to save files and exit applications within the time period that you specified before the actual shutdown.

NOTE: Clients using Windows 9x 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.

For More Information

For more information about `cifs terminate`, refer to the section, "Stopping and Restarting CIFS Sessions," in Chapter 7.

Understanding the Filer Default Configuration

About 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.

Default Exported and Shared Directories

Default Directories Created

When `setup` finishes, two default directories are made available for access by clients. The default directories are:

- the root directory
- the */home* directory

Permissions for the Default Directories

Table 2-5 shows the permissions that are assigned to the default directories when `setup` finishes.

Table 2-5. Permissions for the Default Directories

This directory...	From this client...	Has these permissions...
The root directory	NFS	• full permissions for the root user on the administration host • no permissions for any other user or host • no permissions
	CIFS	• read and write permissions to all files for the Administrator user when logged in to the filer using the root password • no permissions for other users

Table 2-5. Permissions for the Default Directories *(continued)*

This directory...	From this client...	Has these permissions...
The */home* directory	NFS	permissions associated with individual users and with groups through UNIX security database
	CIFS	permissions assigned by the filer administrator for the HOME share

Accessing the Directories

Table 2-6 shows how to access the default directories. Replace *filer* with the host name of your filer.

Table 2-6. Accessing the Directories

To access...	From this client...	Do this...
The root directory	NFS	Mount */filer/vol/vol0*.
	CIFS	Map a drive to *\\filer\C$*.
The */home* directory	NFS	Mount */filer/vol/vol0/home*.
	CIFS	Map a drive to *\\filer\HOME*.
		Or
		Use the Network Neighborhood icon to locate the filer and locate the *HOME* share.

Contents of the etc Directory

The root directory contains an *etc* directory in which the filer configuration files are stored. You can modify the configuration files from the administration host.

Table 2-7 describes the files in the *etc* directory. Note that some of the configuration files might not exist on your filer if you use the filer for CIFS or NFS only.

Table 2-7. Contents of the etc Directory

File name	Contents	File-sharing protocol
.cifs.cat	Domain information (only if the filer is a member of a domain)	CIFS only

Table 2-7. Contents of the etc Directory *(continued)*

File name	Contents	File-sharing protocol
cifsconfig.txt	CIFS commands that the filer used for configuration The filer maintains this file automatically whenever you enter a `cifs` command, which can cause changes you make manually to be lost. To edit this file, terminate CIFS service by using the `cifs terminate` command, edit and save the file, then reboot the filer.	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 Not used if you use NIS to authenticate groups.	CIFS only
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 Not used if you use NIS to authenticate users.	CIFS
rc	Script of commands to be executed when the filer is initializing	All
serialnum	Filer serial number and license codes.	All
shadow	Encrypted password strings and password aging information Not used if you use NIS to authenticate users	CIFS

How The /home Directory Is Used

How the */home* directory is used depends on the file-sharing protocol used by the client. Table 2-8 describes how NFS and CIFS clients use the directory.

Table 2-8. Using the /home Directory

For NFS clients	For CIFS clients
The */home* directory is exported with read and write permissions to all NFS clients, and with root access to the administration host. Clients can mount directories after the filer host name is added to the filer */etc/hosts* file or to the client's name server. The svtx bit (also known as the sticky bit) is set on the */home* directory to prevent users from deleting each other's files. This is a safe configuration for networks that use UNIX permissions to provide security for network-accessible files. You can clear the sticky bit with the `chmod` command.	The */home* directory can be shared by CIFS users as the home share immediately after `setup` finishes. Before users can read and write files in the home share, you must follow the procedures in the section, "Creating a Home Share for Each User," in Chapter 7. By default, CIFS users can write and delete their own files in */home*. For more information about access rights, refer to the section,"Assigning and Changing Access Rights," in Chapter 7.

The /etc/rc File

How the Filer Uses the /etc/rc File

The filer executes the commands in the */etc/rc* file on the root volume at boot time to configure the filer.

If your filer is licensed to run the CIFS protocol, the */etc/rc* file must be present at boot time for the CIFS protocol to be enabled. No CIFS-specific information is entered in */etc/rc* as a result of the `setup` procedure.

All the commands in the */etc/rc* file are executable from the command line—there are no commands that are restricted to being executed from within the */etc/rc* file.

Procedure for Editing the /etc/rc File

To make changes in the filer configuration that take effect every time the filer is booted, perform the following steps:

1. Make a backup copy of the */etc/rc* file.
2. Edit the */etc/rc* file.

NOTE: Do not add CIFS commands in /etc/rc.

See "Editing Configuration Files" for instructions about editing files from NFS and CIFS clients.

3. Save the edited file.
4. Reboot the filer to test the new configuration.

   If the new configuration does not work as desired, repeat steps 2 through 4.

Default /etc/rc File 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 filer
ifconfig e0 'hostname'-0
ifconfig e1 'hostname'-1
route add default MyRouterBox
routed on
options dns.domainname company.com
options dns.enable on
options nis.domainname company.com
options nis.enable on
timezone US/Pacific
savecore
exportfs -a
nfs on
```

Explanation of Default /etc/rc Contents

Table 2-9 explains the commands in the sample */etc/rc* file.

Table 2-9. Default /etc/rc Command Contents

Command	Explanation
`hostname filer`	Sets the filer's host name.

`ifconfig e0 'hostname'-0`
`ifconfig e1 'hostname'-1`
	Sets the IP address for the filer's Ethernet and Gigabit Ethernet interfaces with a default network mask. The arguments in single backquotes expand to "filer" if you specify "filer" 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 specific filer. Refer to "Naming Conventions for Network Interfaces" 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`
	
	The preceding command 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 the section, "Routing," in Chapter 4 for more information about routing.

`options dns.domainname dell.com`
 	
`options dns.enable on`
	These options set the DNS domain.

Table 2-9. Default /etc/rc Command Contents *(continued)*

Command	Explanation
`options nis.domainname dell.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.
`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 panic.
`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.`

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 555-555-1212"
snmp location "ABC corporation, engineering lab"
snmp init 1
```

Naming Conventions for Network Interfaces

Interface Types the Filer Supports

The filer supports the following interface types:

- Ethernet
- Gigabit Ethernet
- Virtual

How Interfaces Are Numbered

Interface numbers are assigned based on the slot in which the interface card is installed. For more information about how network interfaces are numbered, refer to the hardware guide for your specific filer.

How Multiple Ports Are Identified

Some Ethernet interface cards support four ports. These cards are referred to as quad-port interfaces. The filer uses a letter to refer to each port on a quad-port interface. Table 2-10 shows the relationship of port numbers to letters.

Table 2-10. Relationship of Port Numbers to Letters

Port number	Letter
1	a
2	b
3	c
4	d

How Interfaces Are Named

Table 2-11 shows how interfaces other than virtual interfaces are named, and how interface names are combined with card slot numbers and port letters to make their names unique.

Table 2-11. How Interfaces Are Named

Interface type	Letter used in name	Examples of names
Ethernet (single)	e	e0 e1
Ethernet (quad-port)	e	e0a e0b e0c e0d e1a e1b

Virtual Interface Names

The name of a virtual interface is a string that is no longer than 15 characters that meets the following criteria:

- It must begin with a letter.
- It must not contain a space.
- It must not already be in use for a virtual interface.

Virtual interface names are case-sensitive.

About Using Interface Names in Scripts

When you write or modify shell scripts that involve interface names, remember the interface naming conventions that the filer uses. Because the slot in which an interface card is installed is a part of the interface name, different filers might have different interface names.

Filer Host Names

The first time the setup program runs, the filer creates a host name for each installed interface by appending the interface name to the host name.

Host Name Example

A filer named filer with a single Ethernet interface in slot 0 and a quad-port Ethernet interface in slot 1 has the host names shown in Table 2-12.

Table 2-12. Host Name Example

Interface	Host name
Single-port Ethernet card in slot 0	filer-0
Quad-port Ethernet card in slot 1	filer-1a filer-1b filer-1c filer-1d

Reasons to Follow a Special Recovery Procedure

Certain configuration errors can require you to follow a special recovery procedure because

- The filer does not have a local text editor.
- Problems with interface configuration can make the filer inaccessible to clients from which the */etc/rc* file can be edited.

Procedure When the Filer Does Not Boot

If configuration errors prevent the filer from booting from the hard disk, try booting from the diskette. For more information, see the section, "Booting from System Boot Diskette," in Chapter 18.

Procedure When Administration Host Cannot Access the Filer

If your filer becomes inaccessible from the administration host after you change the */etc/rc* file, perform the following steps to recover from the error:

1. Enter commands on the console to configure the interface with the correct address.
2. Enter the `exportfs` command to export the filer root directory to the administration host.
3. Edit the filer */etc/rc* file from the administration host.
4. Reboot the filer.

   If the changes do not correct the problem, repeat Steps 1 through 4.

Core Files

About Core Files

When a hardware or software failure causes the filer to crash, the filer creates a core file that Dell technical support can use to troubleshoot the problem.

Core File Storage in /etc/crash

On the first boot after a system crash, the filer stores the core file in the */etc/crash* directory on the root volume.

What the savecore Command Does

The `savecore` command, which is included in the default */etc/rc* file on the root volume

- Produces a *core.n.nz* file. The *n* in the file name is a number. The string nz indicates that the file is compressed.
- Displays a message on the system console.
- Logs a message in */etc/messages* on the root volume.

Core Dump Space Needed

A core dump file contains the contents of memory and NVRAM. Core dumps are written to a set of areas at the beginning of all the disks. The core dump area on each disk has a fixed size of approximately 20,447,232 bytes. Therefore, a filer or appliance with a large amount of memory can have an insufficient amount of core dump disk space to store a full core dump.

Table 2-13 shows how many disks are needed to store a full core dump for the amount of memory you might have in a specific filer or appliance.

Table 2-13. Core Dump Space

Memory (in MB)	Disks Needed
32	2
64	4
96	5
128	7
160	9
192	10

Table 2-13. Core Dump Space *(continued)*

Memory (in MB)	Disks Needed
224	12
256 or more	14

Message Logging

About Message Logging

The filer maintains messages in the */etc/messages* file on the root volume.

The level of information that the filer records in the */etc/messages* file is configurable.

About the syslogd Daemon and the /etc/syslog.conf File

The message logging daemon, `syslogd`, uses the */etc/syslog.conf* configuration file on the filer root volume to determine how to log system messages.

You can configure `syslogd` to direct system messages to the console, to a file, or to a remote system based on their severity and origin.

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 Format

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

```
facility.level action
```

The facility Parameter

The `facility` parameter specifies the subsystem from which the message originated. Table 2-14 describes the `facility` parameter keywords.

Table 2-14. facility Parameter Keywords

Keyword	Description
`auth`	messages from the authentication system, such as login
`cron`	messages from the internal `cron` facility
`daemon`	messages from filer daemons, such as `rshd`

Table 2-14. facility Parameter Keywords *(continued)*

Keyword	Description
`kern`	messages from the filer kernel
*	messages from all facilities

The level Parameter

The `level` parameter describes the severity of the message. Table 2-15 describes the `level` parameter keywords arranged in order from highest to lowest severity.

Table 2-15. level Parameter Keywords

Keyword	Description
`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 where messages should be sent. Messages for the specified level or higher are sent to the message destination. Table 2-16 describes the possible actions and gives examples of each action.

Table 2-16. action Parameters

Action	Example
Send messages to a file specified by a path.	*/etc/messages*
Send messages to a host name preceded by an @ sign.	*@adminhost*

Table 2-16. action Parameters *(continued)*

Action	Example
Send messages to the console.	*/dev/console* or *

Example Line From /etc/syslog.conf

The following example causes all kernel messages of levels `emerg`, `alert`, `crit`, and `err` to be sent to the */etc/messages* file:

```
err.kern /etc/messages
```

The /etc/messages File Restart Schedule

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.

Checking the /etc/messages File Daily

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.

Sample /etc/syslog.conf File

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

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

Configuring Filer Options

Commands to Use to Set Options

The filer recognizes two commands, `options` and `vol options`, to set options.

"The options Command" describes the syntax for the options command.

"The vol options Command" describes the syntax for the vol options command.

The options Command

What the options Command Does

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

Syntax of the options Command

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

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

If you omit a value for an option, the command displays the current value of the option.

Table 2-17 describes the variables.

Table 2-17. Variables of the options Command

Variable	Description
option	the name of the option
value	the value of the option

Example of the options Command

The following command specifies the recipients of automatic email:

```
options autosupport.to customer@company.com
```

The vol options Command

vol options Command Configures Volume-Level Behavior

You use the vol options command to configure volume-level behavior. You can use this command only with volume options, which are listed in the section, "Volume Options," in Chapter 19.

Changes made with the `vol options` command are persistent between reboots—you do not need to add them to the /etc/rc file.

Syntax of the vol options Command

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

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

If you omit a value for an option, the command displays the current value of the option.

Table 2-18 describes the variables.

Table 2-18. Variables of the vol options Command

Variable	Description
volname	the name of the volume that the option applies to
option	the name of the option
value	the value of the option

Example of the vol options Command

The following command sets the maximum size of a RAID group in the volume named myvol to 12:

```
vol options myvol raidsize 12
```

Sending Automatic Email

How Automatic Email Messages Are Controlled

The filer uses the `autosupport` daemon to control how automatic email messages are sent from your filer to your administrator.

How the autosupport Daemon Works

The `autosupport` daemon is enabled by default on the filer. The daemon triggers automatic email messages to customer-defined administrator accounts, alerting them to potential filer problems.

Mail Host Requirement for autosupport

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 `setup` is used as a mail host. You can specify more mail hosts.

About Configuring autosupport

You can specify up to five addresses of email recipients.

Refer to "Use the Options Command to Configure Autosupport" for more information about specifying the email address and other options.

Events That Trigger autosupport Email

The mail host sends email about your filer after any of the events listed in Table 2-19.

Table 2-19. autosupport Email Trigger Events

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	`POWER_SUPPLY_DEGRADED!!!`
System reboot	`REBOOT`
PowerVault 700N storage system error	`SHELF_FAULT!!!`
Spare disk failure	`SPARE FAIL!!!`
Weekly backup of */etc/messages*	`WEEKLY_LOG`
option `autosupport.doit` `command`	`THE STRING SPECIFIED IN OPTION autosupport.doit`

Contents of Automatic Email Messages

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

- date and time stamp of the message
- Data ONTAP 5.3 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*

Use 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
```

Example: `autosupport.enable off`

Specifying mail hosts: The command to specify hosts that send autosupport email messages 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.

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.

Example: **`options autosupport.to customer@company.com`**

Be sure to enter the command on a single line.

Specifying the Filer Administrator's Address

The `options` command for specifying the filer administrator is as follows:

```
options autosupport.from address
```

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

Sending an Immediate Message

Immediate messages contain the same filer data as automatic messages.

How to send immediate messages: 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.

Example: `options autosupport.doit TESTING`

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 filer by company.com (4.1/SMI-4.1)
id AA14370; Thu, 26 Sep 96 07:51:31 PDT
Message-Id: <9609261451.AA14370@company.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 filer

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.

Filer System Time Synchronization

Commands for Synchronizing Time

The filer uses two commands and a set of options to control and synchronize the system time.

- The `date` command sets the time locally.

  For example, the following command sets the system date and time to 9:25 a.m. on May 22, 1999:

  ```
  date 199905220925
  ```

- The `rdate` command synchronizes the filer's time with the time on another host with an accuracy of one second.
- SNTP (Simple Network Time Protocol) synchronizes the filer's time with the time on a time server with a theoretical maximum accuracy of one nanosecond; actual accuracy depends on the accuracy of the time server and network delays.

Time Synchronization with the rdate Command

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 9x 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.*

When to Use the rdate Command

You use the `rdate` command when you need time accuracy only to the second, or when you want to keep your current method of synchronizing time.

Filer Clock Accuracy

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

```
rdate time_node
```

Use of cron jobs to Run rdate

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.

cron job Example

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

crontab entry: The `crontab` entry on the UNIX system is as follows:

```
0 3 * * * rsh filer -l root rdate time_node
```

Entry in /hosts/equiv file: The */etc/hosts.equiv* file on filer must contain the following line:

```
adminhost adminuser
```

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

Time Synchronization With SNTP

You control how time is synchronized using SNTP and whether time changes are logged by using the `timed` options.

When to Use SNTP

You use SNTP when you need a high degree of accuracy.

List of timed Options

Before using SNTP, you set the `timed` options as described in "Synchronizing Filer System Time" or accept the default settings. The `timed` options are listed in Table 2-20. For details about the `timed` options, see "Timed Options."

Table 2-20. List of timed Options

Timed option	Function	Default
`timed.enable`	Determines whether the timed daemon runs.	Off
`timed.log`	Determines whether to log time changes initiated by the `timed` daemon to the console.	Off
`timed.max_skew`	Sets the maximum allowable discrepancy between filer time and server time.	30m
`timed.proto`	Selects whether to use the protocol used by the `rdate` command or SNTP.	ntp
`timed.sched`	Schedules when to synchronize the time with a time server.	hourly
`timed.servers`	Specifies up to five time servers in order of contact priority.	None

Synchronizing Filer System Time

Description

This procedure describes how to set the `timed` options to synchronize filer system time. You can do this any time. After you complete this procedure, filer time is continually updated automatically.

Prerequisites

You must have available the name of at least one time server. You can get a list of NTP (Network Time Protocol) time servers, which SNTP can use, from *http://www.eecis.udel.edu/~mills/ntp/servers.htm*.

Steps

To set the `timed` options so that you can synchronize filer time, perform the following steps:

1. Turn on the `timed` daemon by entering the following command:

   `options timed.enable on`

2. If desired, turn on console logging of time changes initiated by the `timed` daemon by entering the following command:

   `options timed.log on`

3. Specify the maximum discrepancy allowed before time is not synchronized by entering the following command:

   `options timed.max_skew` ***`nu`***

   `n` is the number of units (specified by *u*).

   `u` is one of the following

   - s for seconds
   - m for minutes
   - h for hours

   The default is 30m, which stands for 30 minutes.

4. Specify a protocol by entering the following command:

   `options timed.proto` ***`protocol`***

   `protocol` is one of the following:

   - rdate for the protocol used by the `rdate` command
   - ntp for SNTP

   The default is ntp.

5. Specify a synchronization schedule by entering the following command:

   `options timed.sched` ***`schedule`***

   `schedule` is one of the following:

   - hourly to synchronize hourly
   - multihourly to synchronize every six hours
   - daily to synchronize every day at midnight
   - a number followed by m to specify an interval of minutes or h to specify an interval of hours

   The default is hourly.

6. Specify up to five time servers by entering the following command:

   `options timed.servers` ***`servers`***

   `servers` is a list of the host names or IP addresses of up to five time servers, separated by commas.

Using options Command Options to Maintain Filer Security

What the Options to the options Command Do

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 non-privileged 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 non-root user changes the owner of a file, the set-user-id and set-group-id bits are cleared. If a non-root 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

About Software Licenses

The filer requires software licenses to enable these services and features:

- NFS
- CIFS
- HTTP
- SnapMirror
- SnapRestore

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 service not previously licensed for your filer.
- The filer's file system becomes corrupt and must be rebuilt.

Dell 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 Services

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

`license` *`protocol=code`*

The *`protocol`* field can be one of these values: `nfs`, `cifs`, `http`, `snapmirror`, and `snaprestore`.

Example: If the license code for NFS is ABCDEFG, enter

```
license nfs=ABCDEFG
```

The events that take place after a `license` command depend on the protocol specified. Table 2-21 discusses the events for each service or feature.

Table 2-21. license Command Service or Feature

Service or feature	Messages

NFS	`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	`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.

HTTP	`http license enabled.`
	`Use "options httpd.enable" to enable http.`
	
	To start HTTP service, enter the following command:
	
	`options httpd.enable on`
	
	Results: 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*.

SnapMirror	`snapmirror license enabled` `snapmirror enabled`
SnapRestore	`snaprestore license enabled` `snaprestore is enabled`

Example: 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:

```
filer>license
nfs=ABCDEFG
cifs=not licensed
http=PQRSTUV
snaprestore=not licensed
```

Disabling a License

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

```
license nfs=DISABLE
```

After you 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 Dell technical support to obtain a copy.

CHAPTER 3
Disk and File System Management

Disk Concepts

Chapter Contents

This chapter covers the following topics:

- Understanding RAID groups
- About disk addresses
- Use disk scrubbing to protect data from media errors
- Understanding hot spare disks
- Understanding hot swap
- Using disks of various sizes
- Understanding usable space on each disk
- Handling disk failures
- Effects of disk failure on filer operation

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.

Multiple RAID groups: 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 PowerVault 700N storage system 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.

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 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.

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 to 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."

About Disk Addresses

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.

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

Fibre channel disk addresses: With fibre channel disks, the disk address is a combination of the disk's adapter number and the disk's fibre channel loop ID. To create the fibre channel loop ID, multiply the shelf ID switch value by eight and add it to the bay number. For example, `ha 8`, `shelf 1, disk 2` has disk ID `8.10`.

For more information: For more information about locating specific drives on PowerVault 700N storage systems, refer to the *Installation and Troubleshooting Guide*.

Use Disk Scrubbing to Protect Data From Media Errors

The filer uses the RAID disk 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 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.

Example: 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
```

Commands to start and stop disk scrubbing: 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
```

Sample messages logged from scrubbing: 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`*`.`

  NOTE: An `Inconsistent parity` *error message might indicate file system corruption. If you get such an error, contact Dell technical support for assistance.*

- 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 must 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 PowerVault 700N storage system while the filer is running, with minimal interruption to a file system. For example, you might want to hot swap a disk into a filer to replace a disk or to add a hot spare disk.

Understanding 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.

Handling Disk Failures

If one block on a data disk fails, the filer uses the parity disk in 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 enables the filer to continue running.

Although the filer can continue to function with a failed disk, if it cannot reconstruct that failed disk on 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.

Effects of Disk Failure on Filer Operation

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 enable 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.

Dell 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.

Command to control 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 data 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 the section, "Displaying Filer Statistics," in Chapter 17.

Volume Concepts

Section Contents

This section covers the following topics:

- Understanding volumes
- Determining the number of volumes to use
- Planning a multiple volume configuration
- Installing a foreign volume

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 includes one 2- to 14-disk volume (a root file system). All remaining disks are spares.

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

The root volume: Each filer must have a *root volume* to boot. The root volume of a filer (configured with either a single volume 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*
- */*

In the */vol/vol0* convention

- */vol*

  Indicates that the next part of the path, such as *vol0* in this example, is a volume name.

- */vol0*

  Indicates the default name of the root volume for a filer. You can change this name using the `vol rename` command.

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*.

Example: 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.

Use of a single volume: 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.

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.

Use of multiple volumes: There are several factors to consider before deciding to create and use 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 other volumes 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

Before you configure a filer with multiple volumes, you must 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.

For instructions about specific volume configuration procedures, see "Volume Management Tasks."

Installing a Foreign Volume

You can remove an entire volume from one filer and install it in another, which makes the moved volume a *foreign* volume to the filer.

Example: For example, you might want to move a volume to a different filer to

- replace the volume's PowerVault 700N storage system with one that has a greater storage capacity
- gain access to the files on a dead filer

When a filer detects a foreign volume at boot time, it places the foreign volume off-line. You can then bring the foreign volume on-line. For more information about installing and bringing up a foreign volume on-line, refer to "Adding a Foreign Volume."

Procedures for Managing Disks and Volumes

Section Contents

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 program, which has a graphical interface.

Disk Management Tasks

About This Section

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 command

`vol create` *`volume`* `-r` *`n`*

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 command

`vol volume options raidsize` *`size`*

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 New Disks

New disks are ones that have never been used. Perform the following steps to install new disks.

1. Install one or more disks according to the refer to the *Installation and Troubleshooting Guide* for your PowerVault 700N storage system.

   The system displays a message confirming that one or more disks were installed, then waits 15 seconds as the disk(s) is turned on. The system recognizes the disks as a hot spare disks.

   If you added multiple disks, they might require 25–40 seconds to come up to speed as the system checks the device addresses on each adapter and returns to normal operation.

2. Type the following command:

   `sysconfig -r`

3. Check the `sysconfig -r` output to verify that the new disk has been added.

Adding Disks to Volumes

To add new disks to a volume, perform the following steps:

1. Enter the command

   `sysconfig -r`

   to verify that there are spare disks available for you to add.

2. Add the disks to a volume by entering the command

   `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 *vol*(1) man page for details about adding disks to volumes.

Removing a Failed Disk

When a RAID disk has failed, you need not enter any commands—just remove the failed disk from the PowerVault 700N storage system.

Removing a Hot Spare Disk

If you want to swap disks because you want to use a hot spare disk in another filer, perform the following steps:

1. Type the following command and use the output to determine the disk number:

   `sysconfig -r`

2. Type the following command to spin down the disk, replacing *disk_name* with the name of the disk from the output in Step 1:

   `disk remove disk_name`

   After the disk stops spinning, the disk is ready to be removed.

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.

Removing an Active File System Disk

If you want to remove a disk because it is logging excessive errors, perform the following steps:

1. Type the following command and use the output to determine the disk number:

   `sysconfig -r`

2. Type the following command to fail the disk, using the disk name from the output in Step 1 in place of *disk_name*:

   `disk fail disk_name`

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

   After the `disk fail` command, the system operates in degraded mode, which means that a disk is missing from the RAID group.

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

Volume Management Tasks

Introduction

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, at the system prompt enter

`vol create newvol n`

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 Creating a New Volume

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

`vol status volume`

To view the RAID group and individual disk information for a particular volume, enter

`vol status -r volume`

To view the RAID group and individual disk information for all volumes, enter

`vol status -r`

Setting Volume Options

To set various volume options, enter

`vol options volume option value`

Converting a Mirror Into a Regular Volume

After you use the SnapMirror™ feature to mirror data to a mirror, you can convert the mirror into a regular volume so that clients can write data to the volume. Enter the following command syntax to convert a mirror to a regular volume:

`vol options volume snapmirrored off`

Although you can convert a mirror into a regular volume, you cannot set the `snapmirrored` option to on to convert a regular volume into a mirror. To start using a volume as a mirror, follow the instructions in Chapter 16, "Data Replication Using SnapMirror," to mirror data to the volume.

CAUTION: After you convert a mirror into a regular volume, the filer stops using it for data replication. If you want to use the volume as a mirror again, you must take the volume off-line and follow the instructions in Chapter 16, "Data Replication Using SnapMirror," to restart the process of replicating data into the volume.

Making a Volume Inactive

To remove a volume from active use upon next reboot, enter

`vol offline` *`volume`*

Reactivating an Off-line Volume

To reactivate an off-line volume, enter

`vol online` *`volume`*

Adding a Foreign Volume

To add a foreign volume, that is, a volume that was previously installed on another filer, you move the disks that contain the volume from the old filer to the destination filer.

To add a foreign volume, perform the following steps:

1. Follow the instructions in the hardware guide to remove the disks from the old filer.
2. Turn off the destination filer and install the disks in the destination filer's PowerVault 700N storage system.
3. Turn on and boot the destination filer.

   Results: When the destination filer boots, it places the foreign volume off-line. If the foreign volume has the same name as an existing volume on the filer, the filer renames it *volume_name(1)*, where *volume_name* is the original name of the volume.

CAUTION: If the foreign volume is incomplete, repeat Steps 1 and 2 to add the missing disks. Do not try to add missing disks while online—doing so will cause them to become hot spare disks.

4. If the filer renamed the foreign volume because of a name conflict, type the following command to rename the volume:

   `vol rename oldname newname`

   Example: The following command renames the volume *vol0(1)* to *vol1*:

   `vol rename vol0(1) vol1`

5. Type the following command to bring the volume on-line in the new filer, replacing *volume_name* with the name of the volume:

   `vol online volume_name`

6. Enter the following command to confirm that the added volume came on-line:

   `vol status`

Destroying a Volume

To destroy a volume, turning its disks back into spare disks, perform the following steps.

1. To deactivate the volume, enter

   `vol offline volume`

2. Enter the **`reboot`** command to reboot the filer.

3. To destroy the volume, enter

   `vol destroy volume`

Renaming a Volume

To rename a volume, perform the following steps

1. Enter

   `vol rename oldvolume newvol`

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

3. Update any CIFS shares that refer to the volume.

Handling Volume Failures

A volume might fail because of 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 system 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.

File Statistics for Volumes

How Data ONTAP 5.3 Provides File Statistics

The `filestats` command provides you with a quick way to display a summary of file statistics within a volume on a filer, by reading file information from a snapshot that you specify.

Information Obtained by the Filestats Command

The output from the `filestats` command provides you with a list containing the following information about files from a snapshot in a volume:

- Size
- Creation time
- Modification time
- Owner

The filestats Command Syntax

The `filestats` command has the following syntax:

`filestats [ages` *ages*`] [sizes` *sizes*`] [timetype {a,m,c,cr}][style {readable,table,html}]volume` *volume_name* `snapshot` *snapshot_name*

The `volume` and `snapshot` arguments are required.

volume_name is the name of the volume.

snapshot_name is the name of the snapshot.

Use the `ages`, `timetype`, `sizes`, and `style` options when you want to list specific file information from a volume. For more information about `filestats` options, see the section "filestats Command Options."

Example With No Options Specified

The following example shows sample output from the `filestats` command, without any options:

```
tpubs-cf2> filestats volume vol0 snapshot hourly.1

VOL=vol0 SNAPSHOT=hourly.1

INODES=274528 COUNTED_INODES=875 TOTAL_BYTES=458354190
TOTAL_KB=143556
```

```
FILE SIZE          CUMULATIVE COUNT       CUMULATIVE TOTAL KB
1K                 465                    1576
10K                832                    3356
100K               853                    3980
1M                 856                    4660
10M                864                    32808
100M               875                    143524
1G                 875                    143254
MAX                875                    143254

AGE(ATIME)         CUMULATIVE COUNT       CUMULATIVE TOTAL KB
0                  0                      0
30D                841                    132780
60D                850                    132932
90D                859                    143464
120D               875                    143528
MAX                875                    143528

UID                COUNT                  TOTAL KB
#0                 873                    143528
#20041             2                      0

GID                COUNT                  TOTAL KB
#0                 851                    41556
#30                21                     1972
#1                 3                      0
```

Use File Statistics for Snapshot Management

File statistics help you determine when to schedule snapshots by enabling you to see when most file activity takes place on a volume. Using the `filestats` command also helps you determine snapshot disk consumption.

Example With Ages Option Specified

Enter the `filestats` command with the `ages` option to display a daily breakdown of file changes in a volume, as shown in the following example:

```
filestats ages 1D,2D,3D,4D,5D,6D,7D,8D,9D,10D,11D,12D,
13D,14D volume vol0 snapshot hourly.0
```

- Use the daily age breakdown displayed in the CUMULATIVE TOTAL KB column of the AGE output to determine the average change in data per day.
- Divide the amount of disk space you want to reserve for snapshots by the daily change average. For example, if you find that the average daily change rate is 3 GB and you have a 200 GB volume, 40 GB (or 20 percent) of which you want to reserve for snapshots, divide 40 by 3 to determine the number of daily snapshots

you can have before exceeding your space limit. In this example, 13 daily snapshots is your limit.

Example to Determine Volume Capacity

You can also use the `filestats` command to determine when most activity occurs on a volume during a given day so that you can effectively schedule hourly snapshots.

The following example shows how you can use the `filestats` command to determine when most file changes occur in a volume within a 24-hour period:

```
filestats ages 1H,2H,3H,4H,5H,6H,7H,8H,9H,10H,11H,12H,
13H,14H,15H,16H,17H,18H,19H,20H,21H,22H,23H,24H volume vol0
snapshot hourly.0
```

If *hourly.0* was taken at 8 a.m., and most file changes took place between 7H and 9H, which corresponds to 3 p.m. and 5 p.m. in this example, you can schedule more snapshots during these hours, and fewer throughout the rest of the day. Scheduling more snapshots before or during increased file activity decreases the time between file changes and snapshots.

For information about managing snapshots, refer to the section, "Managing Snapshot Disk Consumption," in Chapter 9.

Getting a File Statistics Summary

Description

Use the output from the `filestats` command for a summary of the following information about files in a volume:

- Size
- Creation time
- Modification time
- Owner

Restrictions

The following two arguments are required when using the `filestats` command:

- `volume`
- `snapshot`

Step

To use the `filestats` command, enter the following command:

`filestats volume` ***`volume_name`*** **`snapshot`** ***`snapshot_name`*** (where *`volume_name`* is the name of the volume and *`snapshot_name`* is the name of the snapshot).

filestats Command Options

Options to Use With the filestats Command

You can use the following options with the `filestats` command:

- `ages`
- `timetype`
- `sizes`
- `style`
- `expr`

About the Ages Option

The `ages` option of the `filestats` command enables you to see when files have been accessed. You can specify file ages in seconds, hours, and days, using a comma to separate each value. By default, file ages are broken down by days, in 30-day increments.

Example of the Ages Option

For example, to display files with ages under 900 seconds (15 minutes), 4 hours, and seven days, respectively, enter the following command:

```
filestats ages 900,4H,7D volume vol0 snapshot hourly.1
```

The output looks like the following:

```
AGE(ATIME)        CUMULATIVE COUNT        CUMULATIVE TOTAL KB
900               0                       0
4H                0                       0
7D                785                     21568
MAX               882                     146000
```

About the timetype Option

The `timetype` option enables you to specify the time types that you want to list in the age comparison.

Table 3-1 describes the valid `timetype` values you can use with the `timetype` option.

Table 3-1. Valid Values for timetype Option

Value	Definition
`a`	Access time
`m`	Modification time
`c`	File change time (last size/status change)
`cr`	File creation time

About the sizes Option

The `sizes` option enables you to specify the breakdown of sizes, using a comma to separate each value. Default values are in bytes, but you can also use the following three suffixes at the end of a number you specify:

- `K` (kilobytes)
- `M` (megabytes)
- `G` (gigabytes)
- `*` (a special value you use to list all unique file sizes, one line per unique size. Using this command can result in output of several thousands of lines.)

Example of the Sizes Option

For example, to display file sizes in four categories—files with less than 500 kilobytes, files with less than 2 megabytes, files with less than 1 gigabyte, and all other files, enter the following command:

```
filestats sizes 500K,2M,1G volume vol0 snapshot hourly.1
```

The output looks like the following:

```
FILE SIZE        CUMULATIVE COUNT       CUMULATIVE TOTAL KB
500K             862                    4969
2M               866                    10748
1G               882                    146000
MAX              882                    146000
```

About the Style Option

The `style` option controls the output style. The three `style` option arguments are as follows:

- `readable`. This is the default and is what you see when you use the `filestats` command with no `style` option.
- `table`. Use the `table` argument when the `filestats` output will be used by processing programs.
- `html`. Use the `html` argument for output that will be read by a Web browser.

About the expr Option

The `expr` option is an advanced option of the `filestats` command. The `expr` option enables you to specify a Boolean expression that the `filestats` command refers to for each file it encounters.

Table 3-2 lists valid file attributes that you can use with the `expr` option.

Table 3-2. Valid File Attributes for expr Option

Attribute	Definition
`tid`	Tree ID
`type`	File type
`perm`	Permissions
`flags`	Additional flags
`nlink`	Count of hard links
`uid`	Numeric user ID of file owner
`gid`	Numeric group ID of file owner
`size`	Size, in bytes
`blkcnt`	Size, in blocks
`gen`	Generation number
`atime`	Time of last read or write, in seconds
`mtime`	Time of last write, in seconds
`ctime`	Time of last size/status change
`crtime`	Time of file creation
`atimeage`	Access time, in seconds
`mtimeage`	Modification age

Table 3-2. Valid File Attributes for expr Option *(continued)*

Attribute	Definition
`ctimeage`	Size/status time age
`crtimeage`	File creation age

Boolean Expressions to Use With the expr Option

You can use the following standard Boolean expressions as arguments to the `expr` option as shown in Table 3-3.

Table 3-3. expr Option Boolean Expressions

Argument	Argument
&&	>
||	<
==	>=
!=	>=
!	

NOTE: Enclose the entire Boolean string following the `expr` *option in quotation marks. Attributes you use in the string must be enclosed in braces {}.*

Example of the expr Option

For example, if you want to restrict the `filestats` command output to include only files whose size is greater than 10,000 bytes and whose UID is 5150, enter the following command:

```
filestats expr “{size}>10000&&{uid}==5150” volume vol0
snapshot hourly.0
```

Volume Reversion Using SnapRestore

About SnapRestore

The filer’s SnapRestore™ feature enables you to revert a volume to the state it was in when a particular snapshot was taken. Without this feature, you would need to use one of the following methods to restore data to a volume:

- Restoring files from tape
- Copying files from a snapshot to the active file system

Using either of these methods takes a longer time than reverting the volume. This is because with SnapRestore, no data needs to be copied; the file system is just put back in its earlier state.

SnapRestore is a licensed feature. You must purchase and install the license code before you can use it.

How SnapRestore Works

After you select a snapshot for reversion, the filer reboots with the volume containing the same data and timestamps as it did when the snapshot was taken. All data that exists before you initiate the reversion is overwritten.

CAUTION: You cannot undo the reversion to change the volume back to the state it was in before the reversion.

What SnapRestore Reverts

SnapRestore reverts only the file contents. It does not revert attributes of a volume. For example, the snapshot schedule, volume option settings, RAID group size, and maximum number of files remain unchanged after the reversion.

Option settings applicable to the entire filer, however, might be reverted. This is because the option settings are stored in a registry in the */etc* directory. If you revert the root directory, the registry is reverted to the version that was in use at the snapshot creation time. For more information about how reverting a root volume works, refer to "Effects of Reverting a Root Volume."

Files That SnapRestore Cannot Recover

You cannot revert a volume to recover a deleted snapshot. For example, if you delete the *hourly.2* snapshot and revert the volume to the *hourly.1* snapshot, you cannot find the *hourly.2* snapshot after the reversion. Although the *hourly.2* snapshot existed at the creation time of the *hourly.1* snapshot, SnapRestore cannot revert the contents of the *hourly.2* snapshot because you already deleted it.

How SnapRestore Affects Recent SnapShots

After you revert a volume to a particular snapshot, you lose the snapshots that are more recent than the snapshot used for the volume reversion. For example, after you revert the volume to the *hourly.1* snapshot, you no longer have access to more recent snapshots, such as the *hourly.0* snapshot. This is because at the creation time of the *hourly.1* snapshot, the *hourly.0* snapshot did not exist.

Typical Applications of SnapRestore

If a client application corrupts data files in a volume, you can revert the volume to a snapshot taken before the data corruption. The following examples illustrate some situations in which you can apply SnapRestore to recover from corrupted data.

Example: A messaging application or a database application stores user data in one or two files that can grow to several hundred GB in a volume. If, for some reason, this application corrupts the files, you can revert the volume to a snapshot taken before the data corruption.

Example: You can revert a volume used as a test environment to its original state after each test.

Considerations Before Using SnapRestore

This section describes two considerations you must make before deciding whether you should use SnapRestore to revert a volume.

Time required for data recovery: If the amount of corrupted data is small, it is easier to copy the files from a snapshot or restore the files from tape than to use SnapRestore for these reasons:

- You can preserve the data in other files in the same volume.
- The filer does not need to reboot.

If the amount of data to be recovered is large, it takes a long time to copy the files from a snapshot or to restore from tape. In this case, SnapRestore is the preferred method for recovering from data corruption.

Free space required for data recovery: If a file to be recovered needs more space than the amount of free space in the active file system, you cannot copy the file from the snapshot to the active file system. For example, if a 10-GB file is corrupted and only 5 GB of free space exists in the active file system, you cannot copy the file from a snapshot to recover the file. In this case, SnapRestore can quickly recover the file; you do not have to spend time making the additional space available in the active file system.

How SnapRestore Works With SnapMirror

The following list describes how SnapRestore and SnapMirror interact with each other:

- You can revert a volume that is the source volume for data replication. You cannot, however, revert a volume that is currently the mirror for data replication.
- You can revert to any snapshot that is displayed by the `snap list` command. That is, you can revert to a regular snapshot or a snapshot created by SnapMirror for data replication. The snapshots created by SnapMirror have a different naming convention than the regular snapshots, as explained in the section, "Snapshots Created During Data Replication," in Chapter 16.
- If you have both regular and SnapMirror snapshots in the volume, avoid reverting to a snapshot taken before the SnapMirror snapshot. If you must revert to a snapshot taken before the SnapMirror snapshot, after the reversion, the volume contains no SnapMirror snapshot that is essential for the incremental update of the mirror. The filer must re-create the base-line version of the mirror.

- For example, *vol1* is the source volume for data replication and it contains two snapshots: *hourly.0* and *filerA_vol1.2*. If *hourly.0* was taken earlier than *filerA_vol1.2* and you revert *vol1* to *hourly.0*, you cannot find *filerA_vol1.2* after the reversion. As a result, the filer cannot start an incremental update of the mirror. It must re-create the baseline version of the mirror.

Effects of Reverting a Root Volume

Because the */etc* directory of the root volume contains configuration information about the filer, reverting the root volume might change the filer configuration. The following list describes the effects of reverting the root volume:

- The volume loses the changes that were made to the */etc* directory after the snapshot creation time. Suppose you change the IP address of an interface on the filer after the *hourly.0* snapshot was taken. If you revert the root volume to the *hourly.0* snapshot, the filer reboots with the old IP address for the interface.
- The options used for the entire filer are reverted to the settings that were in effect when the snapshot was taken.

Recommendations: Avoid reverting the configuration files. To avoid reverting the configuration files, follow one of these steps:

- Store all data that might need to be reverted in a volume other than the root volume. This ensures that you never need to revert the root volume.
- If the data you want to revert resides in the root volume, back up the */etc* directory to another volume or another filer before using SnapRestore. After you revert the volume, restore the */etc* directory and reboot the filer.

  If you back up the */etc* directory to another volume, you can use the `vol options` *`volume`* `root` command to make the filer reboot with that volume as the root volume. In this way, when the filer reboots during the reversion, it can use the correct settings in the */etc* directory.

Effects of SnapRestore on Filer Backup and Recovery

Because all files in a reverted volume have timestamps that are the same as those when the snapshot was created, the filer's `dump` and `restore` commands might be affected. Incremental backup and restore operations can no longer rely on the timestamps to determine what data needs to be backed up or restored.

Recommendation: After you revert a volume, perform a level-0 backup of the volume. After the level-0 backup is finished, the filer can perform subsequent incremental backups correctly. Also, if you need to restore data from tape to this volume, use only the backups created after the volume reversion.

Reverting a Volume to a Selected SnapShot

Description

Use SnapRestore to revert a volume to a snapshot. You can use this feature at any time. After you enter the command for reverting a volume, the filer reboots with the volume containing the same data as it did when the snapshot was taken.

Prerequisites

You must meet these prerequisites before using SnapRestore:

- SnapRestore is a licensed feature. You must enter the `snaprestore` license code before you can revert a volume to a snapshot.
- Snapshots must exist on the filer so that you can select one snapshot for the reversion.
- The volume to be reverted must be on-line.
- The volume to be reverted must not be a mirror used for data replication.

Cautions

Be sure that you understand the following rules before reverting a volume:

- SnapRestore overwrites all data in the volume. After you use SnapRestore to revert to a selected snapshot, you cannot undo the reversion.
- When you revert a source volume for data replication, try not to select a snapshot taken before the SnapMirror snapshot. If you must revert to a snapshot taken before the SnapMirror snapshot, the filer can no longer perform an incremental update of the mirror; it must recreate the base-line version of the mirror.
- Snapshot deletions are irrevocable. If you delete a snapshot, you cannot recover the snapshot by using SnapRestore.
- After you revert a volume to a selected snapshot, you lose all the snapshots that were taken after the selected snapshot.
- Reverting the root volume causes the filer to reboot with configuration files that were in effect when the snapshot was taken.

Steps

To revert one or more volumes, perform the steps that follow. At any time before you enter **y** in the last step, if for some reason, you do not want to proceed with the volume reversion, enter Ctrl-C.

1. Notify network users that you are going to revert a volume so that they know the current data in the volume will be replaced by that of the selected snapshot. NFS users can unmount the files and directories in the volume before the reversion. If

they do not unmount the files and directories, they might see the "Stale file handle" error message after the volume reversion.

2. If you know the name of the snapshot for reverting the volume, go to Step 5.

   If you want to review the list of snapshots available for volume reversion, enter the following command:

   `vol snaprestore` *`volume`* `...`

   volume is the name of the volume to be reverted. Enter the name only, not the complete path.

   You can enter multiple volume names in the command, separated by spaces.

   Result: The filer displays a warning message and prompts you to confirm your decision for reverting the volume.

   Between the time you enter the `vol snaprestore` command and the time when reversion is completed, the filer stops deleting and creating snapshots.

3. Enter **y** to confirm that you want to revert the volume.

   Result: The filer displays a list of snapshots for you to choose from.

4. Enter the name of the snapshot for reverting the volume and go to Step 7.

   Result: The filer displays the name of the volume and the name of the snapshot for the reversion. Then it asks whether you want to reboot the filer to proceed with the reversion.

5. Enter the following command:

   `vol snaprestore` *`volume`* `-s` *`snapshot`* `[`*`volume`* `-s` *`snapshot`* `...]`

   volume is the name of the volume to be reverted. Enter the name only, not the complete path.

   snapshot is the name of the snapshot for volume reversion.

   You can enter multiple pairs of volume name and snapshot name in the command.

   Result: The filer displays a warning message and prompts you to confirm your decision for reverting the volume.

   Between the time you enter the `vol snaprestore` command and the time when reversion is completed, the filer stops deleting and creating snapshots.

6. Enter **y** to confirm that you want to revert the volume.

   Result: The filer displays the name of the volume and the name of the snapshot for the reversion. Then it asks whether you want to reboot the filer to proceed with the reversion.

7. Enter **y** to confirm that you want to continue with the reversion.

   Result: The filer reboots.

CHAPTER 4
Network Administration

Working With Large Files

About Large Files

You can have large files on your filer. The maximum size of a large file is determined by the smaller of the following two values:

- Maximum file size supported by your server
- Maximum file size supported by your clients

However, if the maximum file size on your client is larger than the maximum file size on your server and you enter a command to display the volume size on your server, the file size is displayed incorrectly. In this case, the maximum file size appears as 2 GB, even though the file on the client is larger than 2 GB.

Software Requirements

To use large files, ensure that your system meets the following requirements:

- Filer operating system: Data ONTAP 5.3
- NFS version 3: enabled
- UNIX host and clients: Solaris 2.6 or later
- Windows NT host and clients; 4.0 or later

How to Enable NFS

To enable NFS version 3, use the `options` command to set the `nfs.v3.enable` option to On.

Using SNMP

About SNMP

You use SNMP (Simple Network Management Protocol) to direct a process, called an agent, on the filer to perform network management tasks such as gathering status and diagnostic information. The information is sent to network management stations, which are client workstations on a network. The network management stations use third-party applications to process the information.

The information that is exchanged to perform these tasks is described in ASCII files called Management Information Bases (MIBs).

Data SNMP Provides

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.

Command to Configure 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 555-555-1212'
snmp location 'ABC corporation, engineering lab'
snmp community add ro private
snmp traphost add snmp-mgr1
snmp init
```

SNMP Commands Supported by Dell

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

- `snmp contact 'jdoe@abc.com 555-555-1212'`

  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 the administrator to contact the right person after receiving an automatic email message.

- `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.

- `snmp traps`

  Generates asynchronous notification of events, called traps.

About the Dell Custom MIB

The Dell custom MIB provides detailed information about many aspects of filer operation. You must use Dell custom MIB 1.1.2 or higher. It contains objects that help you manage multivolume features.

Where to get the MIB: You can obtain the custom MIB from the Data ONTAP 5.3 CD. To locate the MIB on the CD, read the `contents.txt` file.

Installing the MIB: Install the MIB file on your network management workstation according to the installation procedure for your workstation, so that your workstation can obtain information from the filer about the objects that are part of the MIB.

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.

About MIB Group Contents

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

Table 4-1. MIB Group Contents

Group name	Contents
`cifs`	Statistics like those displayed by the `cifs stat` command.
`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.
`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.
`product`	Product-level information, such as the software version string and the system ID.
`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 or Tracking Disk Usage by Using Disk Quotas" in Chapter 11.
`raid`	Information about RAID equivalents to the `sysconfig -r` output. For more information about `sysconfig`, see Chapter 17, "System Information and Performance."
`sysstat`	System-level statistics, such as CPU uptime, idle time, and the number of kilobytes transmitted and received on all network interfaces.

About Traps

You use the `snmp traps` command to inspect the value of MIB variables periodically and send an SNMP trap to the machines on the traphost list whenever that value meets the conditions you specify. The traphost list specifies network management stations that receive trap information.

You use the SNMP third-party applications on your network management station to process the trap information. For example, you can set a trap to monitor the fans on a filer and have the SNMP application on your network management station put a flashing message on your console that tells you that a fan has ceased operating.

You can set traps on any numeric variable in the filer's MIB.

How to Define Traps

You define or change a user-specified trap using the following `snmp traps` command:

`traps` *`trapname.parameter value`*

`trapname` is the name of the trap.

`value` is the value that you assign to the definition.

`parameter` must be a parameter listed in the Table 4-2.

Table 4-2. Parameter Descriptions

Parameter	Description
`var`	A trap's variable is the MIB variable that is queried to determine the trap's value. All MIB variables must be specified in the form `snmp.`*`oid,`* where oid is an OID (Object Identifier). A list of OIDs in the Dell MIB is in the `traps.dat` file in the same directory as the MIB.
`trigger`	A trap's trigger is a piece of code that determines whether the trap should send data. The following triggers are available: `single-edge-trigge`r—sends data when the trap's target MIB variable's value crosses an edge—a value that you specify. `double-edge-trigger`—enables you to have the trap send data when an edge is crossed in either direction (the edges can be different for each direction. `level-trigger`—sends data whenever the trap's value exceeds a certain level.
`edge-1` `edge-2`	A trap's edges are the threshold values that are compared against during evaluation to determine whether to send data.
`edge-1-direction` `edge-2-direction`	Edge-triggered traps only send data when the edges are crossed in one direction. By default this is UP for the first edge and DOWN for the second edge. The direction arguments let you change this default.
`interval`	The interval is the number of seconds between evaluation of the trap. A trap can only send data as often as it is evaluated.

Table 4-2. Parameter Descriptions *(continued)*

Parameter	Description
`interval-offset`	The interval offset is the amount of time in seconds until the first trap evaluation, and is zero by default. You can set it to a non-zero value to prevent too many traps from being evaluated at once (at system startup, for example).
`backoff-calculator`	After a trap sends data, you might not want it to be evaluated so often anymore. For instance, you might want to know within a minute of when a file system is full, but only want to be notified every hour that it is still full. There are two kinds of backoff calculators: stepwise and exponential.
`backoff-step`	The number of seconds to increase the evaluation interval if you are using a step backoff. If a trap's interval is 10 and its backoff-step is 3590, the trap is evaluated every 10 seconds until it sends data, and once an hour thereafter.
`backoff-multiplier`	The value by which to multiply a trap's evaluation interval each time it fires. If you set the backoff calculator to exponential-backoff and the backoff multiplier to 2, the interval doubles each time the trap fires.

It is important to distinguish between the kind of user-specified traps you can set using the `snmp traps` command, and the built-in support for traps, such as coldstart. Built-in traps such as cold start are automatically sent to the hosts on the traphosts list when some event (a reboot in the case of a cold start) occurs. User-specified traps only exist after they are defined by a series of `snmp traps` commands.

Traps are persistent. After you set a trap, it remains across reboots until you specifically remove it.

Host Name Resolution

How the Filer Resolves Host Names

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.

Name Resolution Search

By default, first the filer tries to resolve host names locally by searching the */etc/hosts* file in the root volume and in the */etc/nsswitch.conf* file in the root volume. If it cannot resolve the host name, the filer tries NIS, if NIS is enabled. If the filer still cannot resolve the host name, the filer requests services from a DNS server, if DNS is enabled. You can specify any or all of the resolution methods.

Default Search Order

Table 4-3 shows the default search order for each map.

Table 4-3. Default Search Order for Maps

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

Specifying a Search Order

To specify a different order in which the filer contacts host name services, create the */etc/nsswitch.conf* file in the root volume. Each line must have the following format as described in Table 4-4:

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

Table 4-4. Format Descriptions for a Search Order

Parameter	Description
`map`	One of the following maps or databases: `hosts`, `passwd`, `netgroup`, `group`, or `shadow`.
`service`	One or more of the following:
	files for local files in the */etc* directory in the root volume
	dns for DNS
	nis for NIS

Example Search Order

You can 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 the filer resolves a host, the search stops.

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

Using the /etc/hosts File for Host Name Resolution

The filer can use the */etc/hosts* file in the root volume to resolve host 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, you do not need */etc/hosts*.

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 execute the appropriate `ifconfig` command 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.

Example

You can put the following line at the end of the NIS Makefile section for *hosts.time*:

```
@mntdir=/tmp/dell_etc_mnt_$$$$;\
if [ ! -d $$mntdir ]; then rm -f $$mntdir; mkdir $$mntdir; fi;\
for filer in filer1 filer2 filer3 ; 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 filer in..." list in place of filer1, filer2, and so on.

Using DNS

The filer includes DNS client capabilities to 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.)

NOTE: To prevent naming inconsistencies, Dell recommends that when you enable DNS, you use only the default /etc/hosts file in the root volume.

Enabling DNS during setup: At `setup`, if you enter **y** in response to the following prompt, `setup` 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 for 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 *resolv.conf*(5) man page.

Enabling DNS without using setup: If you didn't start DNS during `setup`, you can start DNS by performing the following steps:

1. Create a */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      .com
   options dns.enable on
   ```

3. Reboot the filer or enter the commands at the filer prompt.

   Result: DNS is now 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, 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 to query NIS servers for host-name-to-IP-address and IP-address-to-host-name translation services.

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.

NIS Maps the Filer Uses

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 setup: During `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 `setup`, you can start NIS by performing the following steps:

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 company.com
   options nis.enable on
   ```

2. Enter the `options` commands from the command line or reboot the filer.

   Result: Options entered only on the command line are not saved if you reboot. NIS is now 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, unless you specify files first in the */etc/nsswitch.conf* file in the root volume.

Disabling NIS: To disable NIS 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.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

About Filer 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. It can, however, route its own packets.

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:

```
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 you must enter explicit information into the filer's routing table, use the `route` command. See the *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 `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 route preference for the router. The highest preference is 1, which is also the default preference for the router specified during `setup`. The lowest preference is 15. For information about the format of the */etc/dgateways* file, refer to "The /etc/dgateways File" on page 13.

Using the routed Daemon to Manage Multiple Routers

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 *route*(1), *routed*(1), and *dgateways*(1) man pages to learn more about routing on the filer.

You turn the `routed` daemon Off and On with the `routed` command.

To turn the `routed` daemon Off, enter

```
routed off
```

To turn the `routed` daemon back On, enter

```
routed on
```

NOTE: 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 as follows:

```
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
```

The `routed status` display shows the following information:

- whether RIP snooping is active (On or Off)
- the current list of default gateways
- the metrics of the default gateways (1 through 15)
- 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) as shown below.

#Gateway	Metric
192.9.200.10	1
eng_gateway	2

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 filer uses the `filer-e1` interface to send packets in response to NFS requests received on the `filer-e1` interface.

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.

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

About the ifconfig Command

The */etc/rc* file in the root volume contains `ifconfig` commands to configure network interfaces, including virtual interfaces, at system boot. You can also manually use the `ifconfig` command when the system is operating.

The ifconfig Command Syntax

The `ifconfig` command syntax is as follows:

```
ifconfig <interface>
    [ [ alias | -alias ] <address> ] [ up | down ]
    [ netmask <mask> ] [ broadcast <address> ]
    [ mtusize <size> ] [ mediatype <type> ]
    [ trusted | untrusted ] [ wins | -wins ]
```

Reasons to Use the ifconfig Command

You use the `ifconfig` command for the following purposes:

- changing the interface's IP address, network mask, or broadcast address
- setting the media type on an Ethernet interface
- setting the maximum transmission unit (MTU)
- configuring the interface up or down

Changing the Interface's IP Address, Network Mask, or Broadcast Address

The following examples show configuring an Ethernet interface and a virtual 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 specific filer. For information about interface naming conventions, refer to "Naming Conventions for Network Interfaces" in Chapter 2.

Setting the Media Type on an Ethernet Interface

The following example shows configuring an Ethernet interface on a filer:

```
ifconfig e1 mediatype tp
```

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 4-5.

Table 4-5. Media Types on an Ethernet Interface

Media type	Description
100tx	100Base-T
100tx-fd	100Base-T, full-duplex
tp	100Base-T
tp-fd	10Base-T, full-duplex
auto	100Base-T/100Base-TX Ethernet using Auto-Negotiation
1000fx	Gigabit Ethernet

Setting the Maximum Transmission Unit (MTU)

The following example shows setting the MTU for an Ethernet interface:

```
ifconfig e0 mtusize 1500
```

Table 4-6 lists the default MTU sizes.

Table 4-6. Default MTU Sizes

Interface	Default MTU size
Ethernet	1500
Gigabit Ethernet	1500

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 following command:

```
netstat -i
```

Configuring the Interface Up or Down

The following example illustrates how to configure interfaces up and down:

```
ifconfig e1a up
ifconfig e0 down
ifconfig virt_interface up
```

Edit /etc/rc File to Make Changes Persistent After Reboot

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

Viewing Interface Configuration Information

Table 4-7 illustrates examples of how to use the `ifconfig` command to view interface configuration information.

Table 4-7. Using the ifconfig Command

Description	Syntax
Show the current configurations of all network interfaces.	`ifconfig -a`
Show the current configuration of a specific network interface.	`ifconfig` *`interface`* **Example:** Enter the following command to show the current configuration of interface e0: `ifconfig e0`

EtherChannel Trunking

Trunks Are a Logical Group of Interfaces

To get the security of failover or the throughput that multiple interfaces working as one interface can provide, you can group up to four Ethernet interfaces. You group them into a logical interface unit known as a trunk. A trunk is composed of links, each of which is an interface. Commands that work on physical interfaces also work on trunks, except that the `ifstat` command returns limited trunk information.

Synonyms for Trunks

Trunks are also referred to by the following terms:

- Virtual aggregations
- Link aggregations
- EtherChannel virtual interfaces.

This document uses the term trunk.

Interfaces Before Trunking

Figure 4-1 shows four separate interfaces, `e3a`, `e3b`, `e3c`, and `e3d`, before trunking.

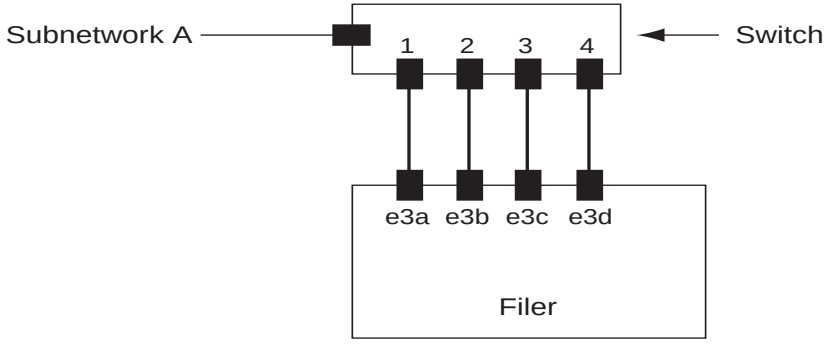


Figure 4-1. Interfaces Before Trunking

Interfaces After Trunking

Figure 4-2 shows the four interfaces after trunking into a multiple-mode trunk called `Trunk1`.

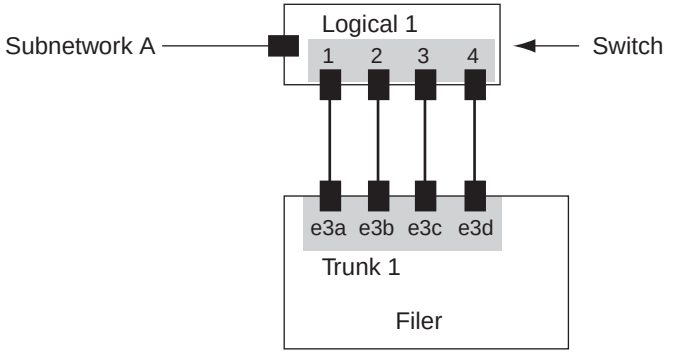


Figure 4-2. Interfaces After Trunking

Kinds of Trunks

Two Kinds of Trunks

There are two kinds of trunks:

- Single-mode trunks enable one link to fail over to another link.
- Multiple-mode trunks enable faster throughput by having links share network loads.

Single-Mode Trunks

In a single-mode trunk, only one of the interfaces is active. The other interfaces are on standby, ready to take over if the active interface fails.

In Figure 4-3, `e0` and `e1` are part of the SingleTrunk1 single-mode trunk. The active interface, `e0`, fails. Failure means that the link status of the interface is down, which signals that the interface has lost connection with the switch. The `e1` interface takes over and maintains the connection. The interface `e1` also takes over the MAC address of the `e2` interface.

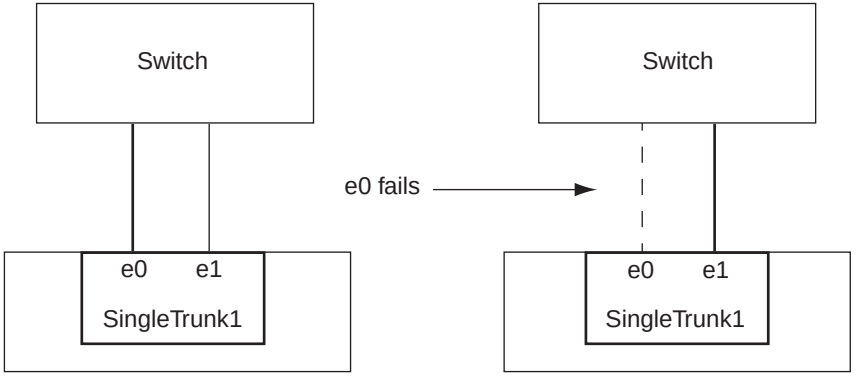


Figure 4-3. Single-Mode Trunks

With single-mode trunks, the filer performs takeover based on the absence of a link.

Multiple-Mode Trunks

In a multiple-mode trunk, all the interfaces are active. This provides greater speed than a single interface.

A multiple-mode trunk requires a switch that supports manually configurable trunking. The switch determines how the load is balanced among the interfaces.

In Figure 4-4, `e0`, `e1`, `e2`, and `e3` are part of the MultiTrunk1multiple-mode trunk. All four interfaces in the MultiTrunk1multiple-mode trunk are active.

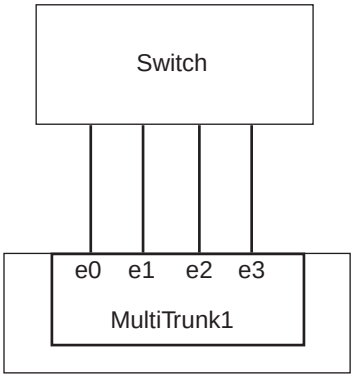


Figure 4-4. Multiple-Mode Trunks

Hardware Requirements for Trunks

To use a multiple-mode trunk, you need a switch that supports manually configurable trunking over multiple port connections. The switch determines how to forward incoming packets to the filer, so you configure the switch so that all the port connections are part of a single logical port. For information about configuring the switch, see the switch documentation.

Filer network interfaces that are part of the same trunk do not have to be on the same network card, but some Ethernet switches and routers require that all members of the trunk be either half duplex or full duplex.

Virtual Interfaces

Trunking Supported by Virtual Interface Feature

The feature that supports trunking is known as a virtual network interface, or virtual interface. The filer treats a virtual interface in the same way as a physical interface, except for the `ifstat` command, which does not show virtual interface information. The `vif` command is customized for virtual interfaces and can provide information that `ifconfig` can't. For example, to get status information about a virtual interface, use the `vif status` command rather than `ifstat`.

Naming Virtual Interfaces

The name of a virtual interface is a string that is no longer than 15 characters that meets the following criteria:

- It must begin with a letter.
- It must not contain a space.
- It must not already be in use for a virtual interface.

Virtual interface names are case-sensitive.

Trunking Virtual Interfaces

You Can Trunk Virtual Interfaces

You create a trunk of virtual interfaces to eliminate a switch as a single point of failure. A trunk of virtual interfaces is known as a second-level virtual interface.

NOTE: With second-level interfaces, if a switch fails and there is a failover to another switch, it might take a few minutes for the spanning tree relay to be reconfigured.

Second-Level Interface Configurations

You can use second-level virtual interfaces on a single filer.

Second-Level Virtual Interfaces on a Single Filer

Why Use Second-Level Virtual Interfaces on a Single Filer

You use second-level virtual interfaces on a single filer to maintain service even if a switch fails.

Example of a Second-Level Virtual Interface on a Single Filer

A subnetwork has two switches that are capable of trunking over multiple port connections. The filer has a two-link multiple-mode trunk to one switch and a two-link multiple-mode trunk to the second switch. You can create a second-level trunking single mode that contains both of the multiple-mode trunks. When the second-level trunk is configured using the `ifconfig`(1) command, only one of the two multiple-mode trunks is brought up as the active link. If all the underlying interfaces in the active trunk fail, the second-level trunk activates the link corresponding to the other trunk.

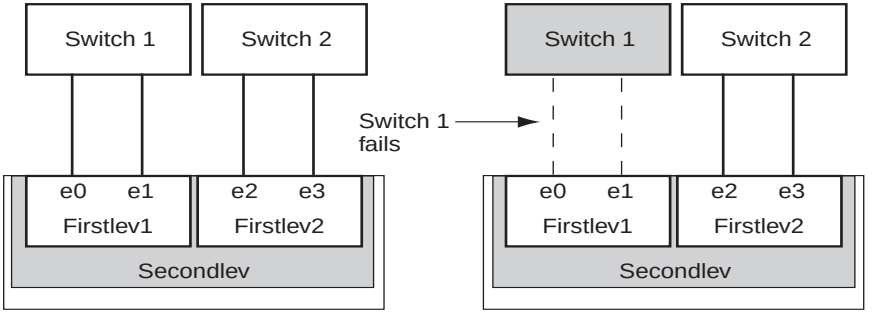


Figure 4-5. Second-Level Virtual Interface on a Single Filer

In Figure 4-5, Secondlev is the second-level virtual interface and is composed of the two virtual interfaces `Firstlev1` and `Firstlev2`. `Firstlev1` is initially the active interface; if Switch 1 drops both links, Switch2 and `Firstlev2` take over and maintain the connection to the network.

Virtual Interface Management

Use the vif Commands to Manage Virtual Interfaces

You manage virtual interfaces with the `vif` command. This command enables you to perform the following actions:

- Create either a single-mode or a multiple-mode trunk.
- Specify a preferred link to activate in a single-mode trunk.
- Add physical interfaces to a trunk.
- Display the status of a virtual interface.
- Display virtual interface statistics, such as the number of packets received and transmitted on each link that makes up a virtual interface. You can specify the time interval, in seconds, at which the statistics are displayed.
- Destroy a trunk.

Put These vif Commands in /etc/rc

The following commands are not persistent:

- `vif add`
- `vif create`
- `vif favor`

Put these commands in the */etc/rc* file to make them persistent across reboots.

Creating a Single-Mode Trunk

Description

Use this procedure to create a trunk in which only one interface is active at a time, thereby ensuring access to a network. After you complete this procedure, the interfaces you specify are combined into a trunk in which one interface is active and the others are ready to take over in case of failure of the active interface.

Prerequisites

You need the following items to complete the procedure:

- A name for the trunk that meets the following criteria:
  - — It must begin with a letter.
  - — It must not contain a space.
  - — It must not already be in use for a virtual interface. Trunk names are case-sensitive.
- A list of the interfaces you want to combine into the trunk.

Step

To create a trunk in which only one interface is active at a time, enter the following command or put it in the */etc/rc* file:

`vif create single` ***`trunk interfaces`*** *(*where ***`trunk`*** is the name of the trunk and ***`interfaces`*** is a list of the interfaces you want the trunk to consist of).

Example

The single-mode trunk shown on "Single-Mode Trunks" was created with the following command:

```
vif creatre single SingleTrunk1 e0 e1
```

Specifying a Preferred Link in a Single-Mode Trunk

Description

Use this procedure to specify an interface you prefer to be active in a single-mode trunk. For example, if you are replacing or upgrading a physical interface and want the new interface to be the active link, you make the new interface the preferred link.

After you complete this procedure, the link you specify is the active link and the other links are not active.

NOTE: If no links are preferred, the active link is selected randomly.

Step

To specify an interface you prefer to be the active link in a single-mode trunk, enter the following command or put it in the in the */etc/rc* file:

`vif favor interface` *(where* **`interface`** is the name of an interface).

Removing a Link From Preferred Status in a Single-Mode Trunk

Description

Use this procedure to remove a link from preferred status in a single-mode trunk. A preferred link has priority for being the active link. You might want to perform this procedure if you are replacing or upgrading a physical interface that is currently the preferred link and want to stop using it.

NOTE: There can only be one preferred link in a trunk.

After you complete this procedure, the currently preferred link is not the preferred link, and you can use the `vif favor` command to make another link the preferred link.

Step

To remove a link from preferred status in a single-mode trunk, enter the following command:

`vif nofavor interface` (where **`interface`** is the name of an interface).

Creating a Multiple-Mode Trunk

Description

Use this procedure to create a trunk in which all interfaces are active at the same time, thereby increasing throughput. After you complete this procedure, you have increased throughput compared to using only a single interface. You can perform this procedure any time you have a configured switch and the physical interfaces available.

Prerequisites

You need the following items to complete the procedure:

- A switch that supports manually configurable trunking configured according to the manufacturer's instructions for a multiple-mode trunk.
- A name for the trunk that meets the following criteria:
  - — It must begin with a letter.
  - — It must not contain a space.
  - — It must not already be in use for a virtual interface. Trunk names are case-sensitive.
- A list of the interfaces you want the trunk to consist of.

Step

To create a trunk in which all interfaces are active at once, enter the following command or put it in the in the */etc/rc* file:

`vif create multi` ***`trunk interfaces`*** (where ***`trunk`*** is the name of the trunk and ***`interfaces`*** is a list of the interfaces that make up the trunk).

Example

The multiple-mode trunk shown in "Single-Mode Trunks" was created with the following command:

```
vif creatre multi MultiTrunk1 e0 e1 e2 e3
```

Creating a Second-Level Virtual Interface on a Single Filer

Description

You use this procedure to create a second-level virtual interface on a single filer. You follow this procedure when you have available two switches configured for multiple-port connections and four or more interfaces. After you complete this procedure, all interfaces are active at once until there is a switch failure, in which case connectivity is maintained by the links to one switch even if the other switch fails.

Steps

To create a second-level virtual interface on a single filer, complete the following steps:

1. Enter the following commands to create two multiple-mode interfaces:

   vif create multi *trunk1 if1 if2*

   vif create multi *trunk2 if3 if4*

   trunk1 and ***trunk2*** are the names of the trunks.

   if1, ***if2***, ***if3***, and ***if4*** are interfaces.

2. Enter the following command to create a single-mode interface from the multiple-mode interfaces:

   vif create single *secondlev trunk1 trunk2*

   secondlev is the name of a second-level virtual interface.

   trunk1 and ***trunk2*** are the names of the trunks.

Example

The following commands create the second-level virtual interface shown in "Example of a Second-Level Virtual Interface on a Single Filer."

```
vif create multi Firstlev1 e0 e1
vif create multi Firstlev2 e2 e3
vif create single Secondlev Firstlev1 Firstlev2
```

Adding Physical Interfaces to a Trunk

Description

Use this procedure to add one or more physical interfaces to a trunk. After you complete this procedure, you get the following results, depending on the type of trunk you are configuring:

- For a single-mode trunk, you get improved reliability by providing additional interfaces to fall back on.
- For a multiple-mode trunk, you get improved throughput.

You can perform this procedure any time after the trunk you are adding interfaces to has been created. You must also configure the switch for the additional ports and physical interfaces available for the additions to take effect.

Step

To add one or more physical interfaces to a trunk, enter the following command or put it in the in the */etc/rc* file:

vif add *trunk interfaces* (where ***trunk*** is the name of a previously configured virtual interface and ***interfaces*** is a list of the physical interfaces you want to add to the trunk).

Displaying the Status of a Trunk

Description

Use this procedure to display the status of a specified trunk. After you complete this procedure, you get information that, for example, is useful in troubleshooting trunk problems. You can use this procedure any time.

Step

To display the status of a trunk, enter the following command:

vif status *trunk* (where ***trunk*** is the name of the trunk. If you don't specify a trunk, the status of all trunks is displayed).

Sample Output

```
vif status
default: transmit 'multi', fail 'log'
vif0: 2 links, transmit 'multi', fail 'default'
    up:
        e3a: state up, since 28Jan1999 23:57:26 (00:00:07)
                mediatype: 100tx-fd
                flags: enabled next-link
                input packets 52, input bytes 3178
                output packets 6, output bytes 553
                up indications 6, broken indications 0
                indication: up at 28Jan1999 23:57:28
                        consecutive 6, transitions 0
        e3b: state up, since 28Jan1999 23:57:24 (00:00:05)
                mediatype: 100tx-fd
                flags: enabled address-set-1
                input packets 34, input bytes 2915
                output packets 5, output bytes 210
```

```
up indications 4, broken indications 0
indication: up at 28Jan1999 23:57:28
        consecutive 4, transitions 0
```

Displaying Trunk Statistics

Description

Use this procedure to display statistics for a specified trunk over a specified period of time. After you complete this procedure, you get information that, for example, is useful in troubleshooting trunk problems. You can use this procedure any time.

Step

To display statistics, enter the following command:

`vif stat` ***`trunk interval`*** (where ***`trunk`*** is the name of the trunk. If you don't specify a trunk, the status of all trunks is displayed. ***`interval`*** is the interval, in seconds. The default is one second.)

Sample Output

This is sample output for the `stat` option.

```
vif stat vif0
Virtual interface (trunk) vif0
    e3a                     e3b
In      Out               In      Out
8637076 47801540          158     159
1617    9588              0        0
1009    5928              0        0
1269    7506              0        0
1293    7632              0        0
920     5388              0        0
1098    6462              0        0
2212    13176             0        0
1315    7776              0        0
```

Destroying a Trunk

Description

Use this procedure to destroy or delete a trunk. You destroy a trunk when you cease needing it or want to use the interfaces for other purposes than a trunk. After you complete this procedure, the interfaces in the trunk act individually rather than as links in a trunk.

Prerequisites

Before you destroy a trunk, you must configure it down using the `ifconfig down` command.

Step

To destroy a trunk, enter the following command:

`vif destroy` ***`trunk`*** (where ***`trunk`*** is the name of the trunk).

Database File Protection

How Data ONTAP 5.3 Provides Database File Protection

Data ONTAP 5.3 provides database file protection through the `nvfail` option of the `vol options` command. The `nvfail` option enables a filer to detect NVRAM inconsistencies at boot time. Use it to warn database administrators of NVRAM problems that could compromise database validity. If the filer finds any problems, database instances hang or shut down, and the filer sends error messages to the console to alert you to check the state of the database.

How to Provide Additional Protection for Database Files

You can provide additional protection to specific database files by adding them to an optional file you create called `/etc/nvfail_rename`. When you enable `nvfail` and the filer detects NVRAM errors at boot up, the filer renames any database files specified in the `nvfail_rename` file by appending `.nvfail` to the original file name. When the filer renames a database file, the database cannot restart automatically. This gives you the opportunity to examine the file for inconsistencies before you remove the `.nvfail` extension and make the file accessible again.

How nvfail Works

When you enable `nvfail`, the following process shown in Table 4-8 takes place during boot-up.

Table 4-8. Enabling the nvfail Option

If...	Then...
The filer detects no NVRAM errors	File service starts normally.
The filer detects NVRAM errors and you use the optional `nvfail_rename` file	a) The filer returns a stale filehandles (ESTALE) error to NFS clients trying to access the database, causing the application to hang, crash, or shut down, and sends an error message to the filer console and log file. b) The filer renames database files specified in the `nvfail_rename` file by appending `.nvfail` to the original file name, making those files unavailable to both CIFS and NFS clients.
The filer detects NVRAM errors and you do not use the optional `nvfail_rename` file	a) The filer returns a stale filehandles (ESTALE) error to NFS clients trying to access the database, causing the application to hang, crash, or shut down, and sends an error message to the filer console and log file. b) No database files are renamed. When the application restarts, files are available to both CIFS and NFS clients, even if you have not verified that they are valid.

Where to Look for Database File Verification Instructions

See the documentation for your specific database software for instructions about examining database file validity.

Error Message Example

When you enable `nvfail` and the filer encounters NVRAM errors, the message sent to the console and the `/etc/messages` file looks like the following:

```
All filehandles have been invalidated due to previous NVRAM
failure.

All filesystems must be remounted by the client(s).
```

If you created an `nvfail_rename` file, you receive an additional message:

```
Renaming files in /etc/nvfail_rename.

nvfail_rename: old_file_name: new_file_name
```

Enabling and Disabling Database File Protection With nvfail

Description

Use the `nvfail` option with the `vol options` command to provide database file protection by turning NVRAM error processing On or Off. The default setting is Off.

Step to Enable nvfail

To enable the `nvfail` option, enter the following command:

`vol options` *`volume_name`* `nvfail on`

volume_name is the name of the volume.

Step to Disable nvfail

To disable the `nvfail` option, enter the following command:

`vol options` *`volume_name`* `nvfail off`

volume_name is the name of the volume.

Using the nvfail_rename File for Additional Database Protection

Description

Use the optional `nvfail_rename` file when you want to rename database files after the filer detects NVRAM errors. This enables you to examine the files for consistency before clients can access them.

Restrictions

You can only list one database file name per line in the `nvfail_rename` file, but you can list as many files as you want.

Steps

To create the `nvfail_rename` file, complete the following steps:

1. Use an editor to create (or modify) the `nvfail_rename` file in the filer's `/etc` directory.

2. List the path name and file name, one file per line, within the `nvfail_rename` file; for example:

   ```
   /vol/vol1/home/dbs/oracle-WG73.dbf
   ```

3. Save the file.

CHAPTER 5
File Sharing Between NFS and CIFS Users

About This Chapter

About File Sharing

This chapter describes how the filer works with NFS and CIFS clients simultaneously. Because these clients interact with a file server differently, you need to understand how the read and write operations performed by one client affect the operations performed by the other client.

File-Locking Interactions

About This Section

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 and NFS users. For details about locking in a particular protocol, consult the documentation for that protocol.

Types of Clients

There are CIFS clients and NFS clients.

Types of Locks

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 (PC)NFS 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 a 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

About Symbolic Links

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.

Controlling Access to Symbolic Links

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, which is the default setting, if the object being accessed by a CIFS client is an absolute or relative symbolic link, 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.

How to Enable and Disable Symbolic Links

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 reenable symbolic links for CIFS, use

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

How to Redirect 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 */u/users/charlie* directory to the */home/charlie* directory on the filer. You do so by specifying symbolic link redirection mappings in a text file named */etc/symlink.translations*.

The format of the */etc/symlink.translations* file 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 */u/users/charlie* directory point to the */home/charlie* directory.

How to Prevent Symbolic Link Cycling

You can 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. 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, which is the 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

About Read-only Bits

The filer, along with the MS-DOS® operating system 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.

How NFS Treats the Read-Only Bit

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 preceding paragraph, a file with the read-only bit turned on 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 superuser.

How the Filer Tracks the NFS or CIFS Client Read-Only Bit

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

About File Naming Conventions

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 Windows operating systems.

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.

How the Filer Generates Short 8.3 File Names

The filer generates an 8.3 file 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*.

Which Clients Support Short File Names

The short names appear on clients that support only the 8.3 format. The short names are not visible to NFS clients. On Windows 9*x* 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*.

Languages and Character Sets

File Names, Languages, and Character Sets

File names in languages other than English can use characters that have diacritic marks, or accents, or can use characters that are not even in the Roman alphabet. UNIX and Windows systems can create file names with no restrictions, provided that they do not violate the naming conventions of the operating system that they were created in. The language you select affects multiprotocol behavior and the code page for Windows 98 and older clients.

File Names Use Character Sets

To make sure that file names stored on the filer are usable by both UNIX and Windows applications, you must choose a character set that contains the characters in the language that your clients use. To do so, you choose a language, and the filer uses a character set that is appropriate to the language.

Every Volume Has a Language

Every volume uses a language, and therefore a character set, that you specify for file names. The root volume determines the code page for PCs and the console character set.

Language Selection

The language you specify controls the name translation between UNIX and Windows names.

NOTE: If a filer is licensed for CIFS, you must set a language for every volume on the filer.

What a Language Applies to

The filer uses a character set appropriate to a given language. The language you select determines which character set the filer uses for the following names:

- User names
- Share names
- System and domain names

NOTE: The following must be in ASCII:

- *Qtree names*
- *Snapshot names*
- *Volume names*

Kinds of Character Sets Supported

The filer supports the following types of character sets. Table 5-1 shows what protocols and operating systems use a particular character set. You use this information to determine what directory format you use.

Table 5-1. Character Sets Supported

Character set type	Description	Used by
ASCII	A 7-bit character set used by most computers. Does not allow letters with accents; that is, diacritics.	NFS, console and log files, qtree names, snapshot names, and volume names.
Unicode	A 16-bit character encoding system. It includes all major languages.	WIN 32 applications and file names in the following systems: • Windows NT • Windows 9*x*
UNIX	A variety of character sets used by UNIX. Can include single-byte and multibyte characters.	NFS, console, and log messages
UTF-8	An ASCII-compatible multibyte Unicode encoding.	Some Solaris clients

Table 5-1. Character Sets Supported *(continued)*

Character set type	Description	Used by
OEM	Single or multiple encodings for older clients	• DOS • Windows 9*x* (except for file names) • NT system and domain name

Languages Supported

The filer supports the languages shown in Table 5-2. The language code for each language appears next to the language. To override the normal UNIX character set so that a UNIX system uses UTF-8 code, add .UTF-8 to the language code.

You use a language code to specify a language with the `vol lang` command, as described in "Setting the Language of a Volume."

Table 5-2. Supported Languages

Language	Language Code
Danish	da
Dutch	nl
English	en
English (US)	en_US
Finnish	fi
French	fr
German	de
Hebrew	he
Italian	it
Japanese euc-j	ja
Japanese PCK(sjis)	ja_JP.PCK
Norwegian	no
Portuguese	pt
POSIX (ASCII only for DOS and ISO-Latin1 for NFS. Equivalent to the Sun locale.)	C

Table 5-2. Supported Languages *(continued)*

Language	Language Code
Spanish	es
Swedish	sv

How to Choose a Language

The flow chart in Figure 5-1 shows how to choose a language.

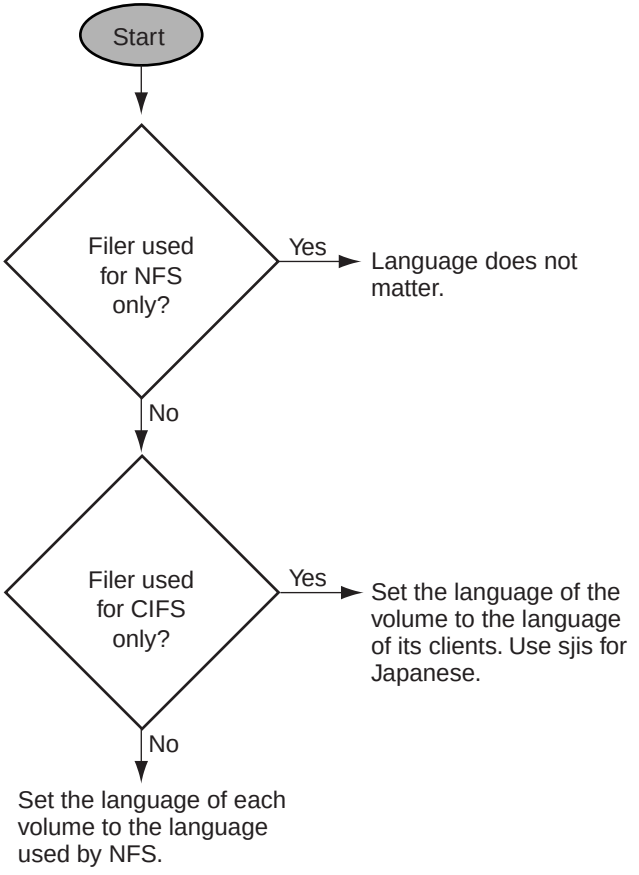


Figure 5-1. Flowchart to Choose a Language

Language Procedures

You can perform the following operations to query and specify the language that a volume uses.

- Show what languages the filer supports.
- Specify a language for the console.
- Change the language, and therefore the character set, that a volume uses for file names.
- Create a volume that uses a specified language other than the default.
- Show each volume with the language that it uses.

Displaying a List of Supported Languages

Description

This procedure displays a list of languages for which the filer supports character sets. You can perform this procedure at any time to determine what language to specify for a volume if you do not already know what languages are supported.

Step

To display a list of the languages for which the filer supports character sets, enter the following command:

```
vol lang
```

Setting the Console Encoding

Description

Use this procedure to set how the console displays non-ASCII information. You would change the encoding if your console does not accept or display characters properly using the current encoding. The default encoding is nfs.

Step

To set the encoding of the filer console, enter the following command:

```
options console.encoding encoding
```

encoding is one of the following encoding styles. You can use both NFS extended (greater than 0x7F) and SGML characters for input.

- nfs—The NFS character set of the root volume. You can use both NFS extended (greater than 0x7F) and SGML characters for input.

- sgml—SGML character format. You can use both NFS extended (greater than 0x7F) and SGML characters for input.
- utf8—UTF-8 character sets. For input, any character greater than 0x7F is treated as the beginning of a UTF-8 encoding.

Setting the Language of a Volume

Description

This procedure sets the language that a volume uses to store file names. You complete this procedure when you want the files names stored in a volume to use a different language than the default language. You should complete this procedure before any files are created in the volume so that all file names use the same language.

After you complete the procedure, the filer stores file names in the selected volume using a character set that best fits the language you select.

Prerequisites

To complete this procedure, you need the language code of the language you want to use. You can get the language code from the table in "Languages Supported."

Caution

Changing the language after the volume contains files can cause some of the NFS encodings to be invalid and is not recommended.

Step

To set the language and character set that a volume uses to store file names, enter the following command:

`vol lang` *`volume langcode`*

`volume` is the name of the volume whose language you want to change.

`langcode` is the code for the language you want the volume to use.

Creating a Volume That Uses a Specified Language

Description

This procedure creates a volume with a specified language. You use this procedure when you are creating a volume but want to use a language different from the

language of the root volume, which is the default language. You use this procedure any time you want to create a volume.

After you complete the procedure, the filer stores file names in the volume you created using a character set that best fits the language you selected.

Prerequisites

To complete this procedure, you need the language code of the language you want to use. You can get the language code from Table 5-2 in "Languages Supported."

Caution

Changing the language of a volume after the volume contains files can cause some of the NFS encodings to be invalid and is not recommended.

Step

To set the language and character set that a volume uses to store file names, enter the following command:

```
vol create volume -l langcode
```

volume is the name of the volume whose language you want to change.

langcode is the code for the language you want the volume to use.

Displaying Which Volume Uses Which Language

Description

This procedure displays a list of volumes with the language each volume is configured to use. You perform this procedure when you want to know which volume uses which language. This is useful in matching clients with languages or deciding whether to create a volume to accommodate clients that use a language that you might not have a volume for.

Step

To display a list of volumes with the language each volume is configured to use, enter the following command:

```
vol status -l
```

Sample Output

Each row of the list displays the name of the volume, the language code, then the language.

```
Volume Language

vol0   ja (Japanese euc-j)
```

CIFS File Name Case

Case Preservation

By default, the filer preserves the case of CIFS file names.

Case Conversion Procedures

You can force all PC-created file names to be stored on the filer in lowercase or return to having the filer preserve case.

To force all CIFS file names to be stored on the filer in lowercase, follow the procedure in "Forcing CIFS File Names to Lowercase." You might want to do this to avoid the following problems:

- UNIX lowercase names might contain uppercase characters when converted to CIFS names.
- Some CIFS clients changing the case of NFS file names on the filer, making them inaccessible from NFS.

To change back to having the filer preserve the case of CIFS file names, follow the procedure in "Preserving the Case of CIFS File Names."

Forcing CIFS File Names to Lowercase

Description

This procedure enables you to force all CIFS file names to be stored on the filer in lowercase. You should do this if you have CIFS clients, so that they do not change the case of NFS file names on the filer that were originally lowercase.

If you perform this procedure, you should do so before Windows files appear on the filer.

After you complete this procedure, the filer stores all CIFS names in lowercase and does not preserve case.

Step

To force all CIFS file names created by CIFS to be lowercase, enter the following command:

```
options cifs.save_case off
```

Preserving the Case of CIFS File Names

Description

This procedure enables you to return from forcing all file names to be stored in lowercase to the default behavior of preserving the case of CIFS file names. You perform this procedure when you want to return to the default behavior.

Caution

File names that were converted to lowercase are not changed.

Step

To return to preserving the case of CIFS file names, enter the following command:

```
options cifs.save_case on
```

Directory Conversion Time

Directory Conversion Can Take a Considerable Amount of Time

Although the conversion process to Unicode directory format is automatic, the initial conversion of a directory can take a considerable amount time, especially if the directory contains a large number of files.

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 or network operations until the conversion is complete.

When There Is no Need to Convert

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.

How to Speed Up Directory Conversion

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

Speeding Up Conversion Time by Renaming NFS Directories

Description

Use this procedure to convert a directory to Unicode format if you have access to a Windows NT client connected to the filer. After you complete this procedure, you have a Unicode directory containing files that were in a non-Unicode directory, and its files are accessible to CIFS clients.

Step

To convert directories to Unicode format quickly, perform the following steps:

1. Create a new CIFS directory from a Windows NT client on the same volume in the same qtree as the directory you want to convert.
2. With the NFS `mv` command, rename files from the directory you want to convert into the directory you just created.
3. Optionally remove the old directory.
4. Optionally rename the new directory.

How to Manage UNIX Access to NTFS Files

UNIX Users Need Windows NT Credentials to Access NTFS Files

If a user is using UNIX and tries to open a file with an ACL in a mixed or NTFS (Windows NT security style) qtree, the filer uses NTFS security semantics to determine whether the user has access to the file.

The filer does this by converting the UNIX UID (User ID) into a Windows NT credential, which is also known as a WAFL (Write Anywhere File Layout) credential. Windows uses the credential to verify that a user has access rights to the file. A UNIX user can have both a UNIX name and a Windows name.

As part of the process of creating a WAFL credential, the filer contacts an NT domain controller to look up a user's SID (Security ID) and groups.

WAFL Credential Caching

Windows verifies each request for access to files. Each access usually takes between 10 and 500 milliseconds, and can take longer because contacting a domain controller to create a WAFL credential is time-consuming. To reduce the time spent in contacting a domain controller, the filer can cache the WAFL credential in a credential cache. This cache is called the WAFL credential cache.

How to Manage the WAFL Credential Cache

The Default Configuration

The default configuration stores each WAFL credential for 20 minutes. This is generally sufficient for most situations.

Two Ways to Manage the WAFL Credential Cache

You can manage the cache globally with options and specifically with the `wcc` command. This enables you to add entries to and remove entries from the WAFL credential cache and to troubleshoot file access problems that might be caused by mapping problems.

NOTE: If you make changes to the WAFL credential cache and an NFS client has cached information, it can take a noticeable amount of time for the changes to become visible.

Global Cache Management Options

If the default configuration does not work well for your site, you can use an option to set how long each WAFL credential cache entry is valid. You can also trace CIFS logins as an aid to debugging mapping problems.

- To set how long each WAFL credential cache entry is valid, use the `wafl.wcc_minutes_valid` option, as described in "Setting How Long Each Wafl Credential Cache Entry Is Valid."
- To display information about every CIFS login attempt, use the CIFS login tracing feature, as described in "Toggling CIFS Login Tracing."

When to Use the wcc Command

You use the `wcc` command to perform the following tasks:

- Add names to or remove names from the WAFL credential cache, as described in "Adding an Entry to the WAFL Credential Cache" and "Deleting Entries from the WAFL Credential Cache.
- Monitor the WAFL credential cache by displaying statistics about it, as described in "Displaying WAFL Credential Cache Statistics."

- Troubleshoot file access and other problems by displaying what name mappings would be, as described in "Displaying a Mapping Result for a UNIX Name" on and "Displaying a Mapping Result for a Windows Name."

The wcc Command Syntax

The `wcc` command has five basic types, each with its own function. The type is determined by the first option, as follows:

- `wcc -a` adds a name to the WAFL credential cache.
- `wcc -d` displays statistics about the WAFL credential cache.
- `wcc -s` displays what the mapping of a Windows NT name would be.
- `wcc -u` displays what the mapping of a UNIX name would be.
- `wcc -x` removes entries from the WAFL credential cache.

The wcc Command Options

Each `wcc` command type has a different set of options, as shown in Table 5-3.

Table 5-3. wcc Command Options

Type	Other options	Function
wcc -a	-u *uname* -i *ip-addr* [-v]	Adds the name to the WAFL credential cache. The `-v` option displays Windows NT groups.
wcc -d	[-v] ...	Displays the following statistics about the WAFL credential cache: • Number of entries in the cache • Age of the oldest entry • Number of Administrator-privileged entries The `-v` option adds mappings for every user.
wcc -s wcc -u	*ntname* [-i *ip-addr*] [-v] *uname* [-i *ip-addr*] [-v]	Displays what the current mapping of the specified name would result in, but does not change the WAFL credential cache. The `-v` option displays numeric SIDs.

Table 5-3. wcc Command Options *(continued)*

Type	Other options	Function
wcc -x	[-f] [-v]	Removes all entries from the WAFL credential cache. The `-v` option displays how many entries have been removed.
	-i *ip-addr* [-v]	Removes all entries from the WAFL credential cache with the same IP address. The `-v` option displays how many entries have been removed.
	-s *ntname* [-i *ip-addr*] [-v]	Removes the entries with the given Windows NT name and optional qualifying IP address. If the Windows NT name is the name of a group, removes all members of that group from the WAFL credential cache. The `-v` option displays how many entries have been removed.
	-u *uname* [-i *ip-addr*] [-v]	Removes the entries with the given UNIX name and optional qualifying IP address. The `-v` option displays how many entries have been removed.

The `-v` option increases the level of detail of information. The kind of information varies with each command. You can have up to three instances of the -v option (-vvv) per command. Each repetition of the option increases the level of detail; three instances provide statistics that are only of interest to Dell technical support.

Setting How Long Each WAFL Credential Cache Entry Is Valid

Description

This procedure enables you to change how long each WAFL credential cache entry is valid. If you need to see security updates as they occur, you might want to use a smaller value than the default. However, access verifications are more frequent than with a greater value. This means that with a smaller value, users might experience slower performance.

Step

To change how long each WAFL credential cache entry is valid, enter the following command:

```
options wafl.wcc_minutes_valid n
```

n is the number of minutes you want each entry to be valid. It can range from 1 through 20160. The default value is 20.

Adding An Entry to the WAFL Credential Cache

Description

This procedure adds an entry to the WAFL credential cache. You can use this procedure in a script to load the WAFL credential cache at boot time with entries rather than wait for those entries to be created in the course of accessing a file. You can perform this procedure at any time.

Prerequisites

You must have the names and IP addresses of those you want to add to the WAFL credential cache.

Cautions

If you add more entries than the maximum number of entries allowed, the older entries are deleted.

Step

To add an entry to the WAFL credential cache, enter the following command:

```
wcc -a -u uname -i ipaddress
```

uname is the UNIX name of a user.

ipaddress is the IP address of the host that the user is on.

Deleting Entries From the WAFL Credential Cache

Description

This procedure deletes entries from the WAFL credential cache. You do this to force the lookup of UIDs the next time they are used, but you don't want to wait until the entries time out automatically.

Prerequisites

You must have the name and optionally the IP address of the person or group you want to remove from the WAFL credential cache.

Caution

If the Windows NT name is the name of a group, this procedure removes all members of that group from the WAFL credential cache.

Step

To remove an entry from the WAFL credential cache, enter the following command:

```
wcc -x name
```

name can be one of the following specifications:

- `-s` followed by a Windows user name or group name
- `-u` followed by a UNIX name

You can further narrow the specification of a user by adding `-i`, followed by the IP address of the host that the user is on.

Displaying WAFL Credential Cache Statistics

Description

This procedure displays WAFL credential cache statistics in detail. You can do this at any time to monitor the WAFL credential cache.

Step

To display statistics about the WAFL credential cache, enter the following command:

```
wcc -d
```

You can get more detailed information by appending `-v` to the command line.

Sample Output

The following sample output shows the output of statistics with the `-d` option.

```
wcc -d

mday (UID 10050) from 10.100.4.41 => NT-DOMAIN\mday*


Total WCC entries: 3; oldest is 127 sec.

Total Administrator-privileged entries: 1

 * indicates members of "BUILTIN\Administrators" group
```

The following sample output shows the output of statistics with the `-v` option used twice.

```
cc -dvv

mmm (UID 1321) from 10.100.4.41 => NT-DOMAIN\mmm

        ***************

        UNIX uid = 1321


        NT membership

                NT-DOMAIN\mmm

                NT-DOMAIN\Domain Users

                NT-DOMAIN\SU Users

                NT-DOMAIN\Installers

                NT-DOMAIN\tglob

                NT-DOMAIN\Engineering

                BUILTIN\Users

        User is also a member of Everyone, Network Users,

        Authenticated Users

        ***************

mday (UID 10050) from 10.100.4.41 => NT-DOMAIN\mday*

        ***************

        UNIX uid = 10050
```

```
        NT membership
                NT-DOMAIN\mday
                NT-DOMAIN\Domain Users
                NT-DOMAIN\Domain Admins
                NT-DOMAIN\SU Users
                NT-DOMAIN\Installers
                BUILTIN\Users
                BUILTIN\Administrators
        User is also a member of Everyone, Network Users,
        Authenticated Users
        ***************
hawleyr (UID 1129) from 10.100.4.41 => NT-DOMAIN\hawleyr
        ***************
        UNIX uid = 1129

        NT membership
                NT-DOMAIN\hawleyr
                NT-DOMAIN\Domain Users
                NT-DOMAIN\Installers
                NT-DOMAIN\SU Users
                BUILTIN\Users
        User is also a member of Everyone, Network Users,
        Authenticated Users
        ***************

Total WCC entries: 3; oldest is 156 sec.
Total Administrator-privileged entries: 1
 * indicates members of "BUILTIN\Administrators" group
```

Displaying a Mapping Result for a UNIX Name

Description

This procedure displays what the current mapping of the specified UNIX name of a UNIX user would result in, but does not change the WAFL credential cache itself. You use this procedure if a UNIX user cannot access a file that the user should be able to access, and you suspect that mapping problems might be part of the problem.

Step

To display what the current mapping of a UNIX name would result in, but not change the WAFL credential cache, enter the following command:

```
wcc -u uname
```

uname is the UNIX name of a user.

You can further narrow the specification of the user by adding `-i`, followed by the IP address of the host that the user is on.

You can get more detailed information by appending `-v` to the command line.

Sample Output

The following example shows the mapping of a UNIX name.

```
wcc -u fuser001
(NT - UNIX) account name(s):  (NT-DOMAIN\fuser001 - fuser001)
        ***************
        UNIX uid = 1172


        NT membership
                NT-DOMAIN\fuser001
                NT-DOMAIN\Domain Users
                BUILTIN\Users
                TFILER\Test
        User is also a member of Everyone, Network Users,
        Authenticated Users
        ***************
```

Displaying a Mapping Result for a Windows Name

Description

This procedure displays what the current mapping of the specified UNIX name of a Windows NT account would result in, but does not change the WAFL credential cache itself. You use this procedure if a Windows NT user cannot access a file that the user should be able to access, and you suspect that mapping problems might be part of the problem.

Step

To display what the current mapping of a Windows name would result in, but not change the WAFL credential cache, enter the following command:

```
wcc -s uname
```

uname is a Wndows NT account.

- You can further narrow the specification of the user by adding `-i`, followed by the IP address of the host that the user is on.
- You can get more detailed information by appending `-v` to the command line.

Sample Output

The following example shows the mapping of a Windows name.

```
wcc -s bluebottle

(NT - UNIX) account name(s):  (NT-DOMAIN\bluebottle - pcuser)

        ***************

        UNIX uid = 65534


        NT membership

                NT-DOMAIN\bluebottle

                NT-DOMAIN\Domain Users

                BUILTIN\Users

        User is also a member of Everyone, Network Users,

        Authenticated Users

        ***************
```

Toggling CIFS Login Tracing

Description

You can enable login information to be displayed after every CIFS login. You can do this any time you need to troubleshoot mapping problems. Login problems are often the cause of users being denied access to files they should have access to. CIFS login tracing is especially useful in producing verbose messages when a login attempt fails.

Caution

Turning on CIFS login tracing should be used carefully because every CIFS login is traced and results in console messages. Constant use of CIFS login tracing can result in many console and log messages.

Step to Turn On CIFS Login Tracing

To turn on CIFS login tracing, enter the following command:

```
options cifs.trace_login on
```

Sample Output

```
If user jdoe attempts to log in, messages like the following
appear on the console:

Mon Jan  4 15:21:38 PST [CIFSAuthen]: Login attempt by DELL\jdoe
from SMITH-PC

Mon Jan  4 15:21:38 PST [CIFSAuthen]: User authenticated by DC

Mon Jan  4 15:21:38 PST [CIFSAuthen]: PC user name maps to UNIX
user smith

Mon Jan  4 15:21:38 PST [CIFSAuthen]: Unix user set to root by
wafl.nt_admin_priv_map_to_root

Mon Jan  4 15:21:38 PST [CIFSAuthen]: Login accepted
```

Step to Turn Off CIFS Login Tracing

To turn off CIFS login tracing, enter the following command:

```
options cifs.trace_login off
```

CHAPTER 6
NFS Administration

This chapter has the following three sections:

- Managing NFS Exports
- Configuring a Filer for WebNFS
- Displaying NFS Statistics

Managing NFS Exports

Introducing the /etc/exports File

/etc/exports Controls Client Access to Directories

The */etc/exports* file controls how NFS clients access filer directories. You add entries to the */etc/exports* file for all the directories you want to export.

Format for /etc/exports Entries

The format of the entries in the */etc/exports* file is as follows:

```
filer_directory_path export_specification
```

Filer Directory Path Format

The filer directory path specifies which directory is made available to clients. The format is as follows:

`/vol/`*`volume_name/directory`*

Examples: The following lines show examples of filer directory paths:

```
/vol/vol0/home
/vol/users/local/jarnold
```

Export Specification Determines Access Privileges

The export specification specifies the privileges that clients have to mount and access the filer directory path. The format is as follows:

`-root=`*`list`*`,-access=`*`list`*`,-ro=`*`list`*`,-rw=`*`list`*

One keyword is required

The keywords `-root`, `-access`, `-ro`, and `-rw` are all optional; however, you must include at least one keyword in an export entry.

What the list variable represents

The *list* variable represents a list that includes one or more

- host names
- netgroup names
- subnets

You can combine elements in an entry

You can combine host names, netgroup names, and subnets in an entry, as shown in the following example:

```
-root=adminhost:administrators,-rw=blender:pcusers:123.45.67.0/
24
```

Example 1: exporting default filer volume to administration host

The following line exports the root directory of the default filer volume to the administration host with root privileges. The administration host can mount the directory, change permissions and ownerships, and create and delete directories and files.

```
/vol/vol0 -root=adminhost
```

Example 2: exporting home directory to administration host and clients

The following line exports the home directory from the default filer volume to the administration host and two clients. The administration host has root privileges and can mount the directory, change permissions and ownerships, and create and delete directories and files. The clients can mount the directory and create and delete directories.

```
/vol/vol0/home -root=adminhost,-rw=blender:mixer
```

Rules For Exporting Volumes And Directories

Export Each Volume Separately

If the filer has multiple volumes, you must export each volume separately; you cannot export all volumes by specifying `/vol` as the exported directory.

Example

The following lines show how to export the volumes on a filer that has three volumes:

```
/vol/vol0 -root=filer, -access=mixer:blender
/vol/users -root=filer, -access=mixer:blender
/vol/builds -root=filer, -access=mixer:blender
```

Nonexample

The following line shows an invalid entry for the */etc/exports* directory for a filer that has three volumes; */vol* cannot be used by itself as a path:

```
/vol -root=filer, -access=blender:mixer
```

Filer Must Resolve Host Names

To export directories to hosts, the filer must be able to resolve host names into IP addresses. Host name resolution can take place

- through DNS name resolution
- by using the */etc/hosts* file on the filer root volume

Refer to "Host Name Resolution" in Chapter 4 for more information about host-name-to-IP-address resolution.

Cannot Restrict Access By Host

Neither the filer nor NFS provide a way to specify hosts that *cannot* mount the filer. To restrict access to exports, you must export volumes and directories in a manner that specifically includes those hosts that should be allowed to mount them. Exclusion occurs when a host is not specifically included in the list of hosts authorized to mount exported volumes and directories.

You Can Export Ancestors and Descendants

The filer permits directories that have exported ancestors to be exported. In many implementations of the UNIX operating system, you cannot export a directory that has an exported ancestor in the same file system.

Example

The following lines show exports that the filer allows:

```
/vol/vol0 -access=adminhost,-root=adminhost
/vol/vol0/home -access=blender:mixer
```

Nonexample

The following lines show exports that are not allowed on some UNIX systems:

```
/home -root=adminhost,-access=blender:mixer
/home/local -root=adminhost,-access=blender:mixer
```

Filer Determines Permissions by Matching Longest Prefix

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.

NOTE: Because of the way the filer determines permissions, it makes little sense to give a client greater permissions at a higher level in the file system.

Example

The following lines show an */etc/exports* file that creates a security breech by enabling any host to mount the */vol/vol0* directory while restricting specific hosts to mounting the */vol/vol0/home* directory. In this example, any host can gain access to the */vol/vol0/home* directory by mounting the */vol/vol0* directory:

```
/vol/vol0
/vol/vol0/home -access=red:blue:green
```

Edit /etc/exports After Changing Volume Names

If you rename a volume using the `vol rename` command

- Entries in the */etc/exports* file that refer to the volume become incorrect.
- The in-memory information about active exports gets updated automatically, and clients continue to access the exports without problems.
- The filer displays an error message when the `exportfs -a` command is entered, or when the filer is rebooted.
- Clients display the error message, "Stale NFS file handle," after the filer is rebooted.

CAUTION: To ensure that the entries in the */etc/exports* file remain valid, always edit the entries in the file to reflect volume name changes immediately after renaming volumes.

Default /etc/exports Entries

/vol/vol0 and /vol/vol0 home Are Exported by Default

By default, the root volume (*/vol/vol0*) and the */vol/vol0/home* directory are exported to the administration host when you run `setup`.

Example of Default exportfs File

The default */etc/exports* file contains the following lines; in this example, the name of the administration host is silver:

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

The contents of the default */etc/exports* file is shown below:

- /vol/vol0 -access=adminhost, -root=adminhost

  Only the administration host (adminhost) can mount the root directory and modify files in the directory.

NOTE: On filers with a single volume, you can refer to the root directory without the /vol prefix.

- /vol/vol0/home -root=adminhost

  The administration host (adminhost) can mount the home directory as root.

  All other clients can access the home directory to read and write files.

Restricting Access to Volumes and Directories

Use Export Options to Restrict Directory Access

You can use the export options to restrict access to directories in various ways.

Restricting Access to /home

You can restrict access to the */home* directory to particular groups by

- using the `chmod` command to change access modes for the directory
- using the `-rw` or `-access` options in the */etc/exports* file to limit write privilege to specific hosts

The -access Option

The `-access` option lists the hosts that can mount exported directories. When you use the `-access` option, only the hosts listed can mount the associated directory.

Syntax

The syntax for the `-access` option is as follows:

`-access=`*`hostname`*`[:...:`*`hostname`*`]`

Limits

There is no limit to the number of host names you can specify with the `-access` option. However, the length of a line in the */etc/exports* file cannot exceed 1,024 characters.

NOTE: If you cannot fit all the host names in a 1,024-character line, you can use net-groups in place of host names.

The -root Option

The `-root` option lists the hosts that can mount exported directories as root. Hosts that mount exported directories as root have full control over the directories and can perform the following operations:

- Create and delete directories and files.
- Change ownership and group associations of directories and files.
- Set access permissions for directories and files.

Syntax

The syntax for the `-root` option is as follows:

`-root=`*`hostname`*`[:...:`*`hostname`*`]`

Limits

You can specify 1 to 256 host names with the `-root` option.

Restrictions

You cannot use netgroup names with the `-root` option.

The -rw Option

The `-rw` option lists the hosts that can modify the exported directories; hosts not listed by the `-rw` option have read privilege only.

Syntax

The syntax for the `-rw` option is as follows:

`-rw=`*`hostname`*`[:...:`*`hostname`*`]`

Limits

You can specify 1 to 256 host names with the `-rw` option.

Restrictions

You cannot use netgroup names with the `-rw` option.

The -ro Option

The `-ro` option lists the clients that cannot modify the exported directories.

Syntax

The syntax for the `-ro` option is as follows:

```
-ro=hostname[:...:hostname]
```

The exportfs Command

Using the exportfs Command

You use the `exportfs` command to export and unexport volumes and directories. Depending on which options you use, the command exports the volumes and directories listed in the */etc/exports* file or a specific volume or directory.

Syntax

The syntax for the `exportfs` command is as follows:

`exportfs [ -aiuv ] [ -o options ][ `*`filer_directory_path`*]

Table 6-1 describes the options.

Table 6-1. exportfs Command Options Syntax

Option	Description
`-a`	Exports all the entries listed in the */etc/exports* file.
`-u`	Unexports all the entries listed in the */etc/exports* file. *NOTE: When you use the* `-u` *option with the* `-a` *option, all exports are unexported regardless of whether they were created from the /etc/exports file or with the* `-o` *option.*
`-i`	Exports all the export entries listed in the */etc/exports* file, but ignores the options specified for the entries.
`-v`	Prints each path name as it is exported or unexported.
`-o`	Specifies the options for a volume or directory that you include in the command line. **Example:** To export the */vol/vol/home/terry* directory to the host "mixer" with read/write access, you enter the following command: `exportfs -o -rw=mixer /vol/vol0/home/terry` *NOTE: Volumes and directories that are exported directly rather than through the /etc/exports file remain exported until canceled using the* `-au` *option or until the filer is rebooted.*

Canceling All Exports

To cancel all exports, enter the following command:

```
exportfs -au
```

Updating Exports Through /etc/exports

When you make changes to the */etc/exports* file, take one of the following actions to make the changes take effect:

- Run the `exportfs -a` command.

Or

- Reboot the filer.

If you delete entries from the */etc/exports* file when you make changes, take one of the following actions to activate the changes and ensure that deleted exports are deactivated:

- Run the following commands:

```
exportfs -au
exportfs -a
```

Or

- Reboot the filer.

The /etc/netgroup File

The /etc/netgroup File Defines Groups of Clients

The filer */etc/netgroup* file defines groups of clients that the filer uses for checking access permission while processing a mount request.

Syntax

The following line shows the syntax for each line in the */etc/netgroup* file:

```
groupname member-list
```

Limits

Each line in the */etc/netgroup* file is limited to 4,096 characters.

Member-list syntax

Each element in `member-list` is

- another group name
- an entry in the following form:

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

  — An element in an entry can be blank, but the commas must be present.

  — When group names are used in the */etc/exports* file, the `username` and `domainname` fields are ignored.

  — When domain names are used, they must be DNS names; NIS names cannot be used.

Restrictions

You cannot use netgroup names with the `-rw` and `-root` options.

Changes Take Effect Immediately

Changes made to the netgroup file take effect immediately.

Example of /etc/netgroups

The following lines show an example of a */etc/netgroups* file:

```
trusted-hosts (adminhost,,)
untrusted-hosts (red,,) (blue,,) (green,,)
all-hosts trusted-hosts untrusted-hosts
```

Example of /etc/exports Using Netgroups

The following lines show an example of an */etc/exports* file that uses netgroup group names:

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

Copy /etc/netgroup When Filer Doesn't Use NIS

If your filer is not configured as an NIS client, the network groups on the filer are not linked with NIS.

Must copy NIS netgroup file

You must copy an existing NIS network group over to */vol/vol0/etc/netgroup* on the filer before the `exportfs` command can use it.

Automating copying with a Makefile

You can modify the Makefile of the NIS master to copy the NIS master's */etc/netgroup* file to the filer when it is changed.

Example Makefile

The following lines of code in the NIS Makefile section for `netgroup.time` copy the */etc/netgroup* file to filers named filer1, filer2, and filer3; substitute the name of your filers in the "for" list, in place of *filer1*, *filer2*, and *filer3*, and add any other filer names to which you want the file copied:

```
@mntdir=/tmp/nac_etc_mnt_$$$$;\
if [ ! -d $$mntdir ]; then rm -f $$mntdir; mkdir $$mntdir; fi;\
for filer in filer1 filer2 filer3 ; 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

About Exporting to Subnets

You can export a directory to clients on a subnet rather than to individual clients.

Valid Export Options for Subnets

The valid export options in the */etc/exports* file for exporting to subnets are as follows:

`-ro=`*`subnet_address`*`[:`*`subnet_address`*`]...`

`-rw=`*`subnet_address`*`[:`*`subnet_address`*`]...`

`-root=`*`subnet_address`*`[:`*`subnet_address`*`]...`

Format for IP Subnet Addresses

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

`dotted_ip/num_bits`

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
  - 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*.

Export to a Subnet as You Do to a Client

You export a directory to a subnet as you do to an individual client, except that you specify a subnet address rather than a full IP address in an export option.

Example:root access:

To export */vol/vol0/home* on the filer for root access to a client named silver and all addresses of the form 123.45.67.x with a netmask 255.255.255.0 (24 leading bits), place the following entry in the */etc/exports* file:

```
/vol/vol0/home -root=silver:123.45.67.0/24
```

Example 2: read/write access

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), place the following entry in the */etc/exports* file:

```
/vol/vol0/home -rw=123.45.0.0/16
```

Example 3: equivalent methods for exporting

The following entries in the */etc/exports* file are equivalent. They export */vol/vol0/home* to a client named host1, the specified subnet, and a client named host2.

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

Configuring a Filer for WebNFS

About Configuring a Filer for WebNFS

The Filer Can Respond to NFS Requests From Browsers

The filer can use NFS rather than HTTP to respond to file transfer requests made through Web browsers that support the WebNFS protocol.

The filer does not need a license for the HTTP protocol to respond to WebNFS requests; however, the filer must be licensed for the NFS protocol.

Web Browser Requirements

To access files through the WebNFS protocol, users type URLs that start with "nfs://". Web browsers must be capable of sending requests using the WebNFS protocol.

Advantages of WebNFS

With WebNFS, the filer can transfer files much faster than with HTTP because the WebNFS protocol can transfer several files, including graphics files, with only one TCP connection. The HTTP protocol, in version 1.1, requires a separate connection for each file that is transferred.

How WebNFS Restricts File Access

WebNFS access does not use the `mount` command to enable access to a subtree, and does not consider UID/GID mappings.

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.

Setting Up WebNFS

Procedure for Setting Up WebNFS

To set up WebNFS, perform the following steps:

1. Enter the following command to turn on WebNFS:

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

2. If you...

   Want to specify a public directory, known as the root directory, for WebNFS access

   Then...

   Enter the following commands, replacing *directory* with the path to the root directory:

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

   ```
   options nfs.webnfs.rootdir.set TRUE
   ```

   Results: All NFS lookups are done relative to the root directory. All WebNFS clients can access files and directories under the root directory.

   Do not want to specify a public directory for WebNFS access

   Do nothing.

Example of Specifying WebNFS Root Directory

To use the */vol/vol0/webfiles* directory as the WebNFS root directory, you enter the following commands:

```
options nfs.webnfs.rootdir /vol/vol0/webfiles
options nfs.webnfs.rootdir.set TRUE
```

Managing WebNFS

Tasks You Can Perform

You can perform the following tasks to manage WebNFS service:

- Change the root directory.
- Disable the root directory.
- Turn off WebNFS service.

Changing the Root Directory

To change the WebNFS root directory, perform the following steps:

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

1. Enter the following command, replacing *newdir* with the path of the new root directory:

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

   Example: To change the root directory to */vol/vol0/corpwebfiles*, you the following command:

   ```
   options web.webnfs.rootdir /vol/vol0/corpwebfiles
   ```

2. Enter the following command to enable the root directory:

   ```
   options nfs.webnfs.rootdir.set TRUE
   ```

Disabling the Root Directory

To disable the root directory, enter the following command:

```
options nfs.webnfs.rootdir.set FALSE
```

Turning Off WebNFS Service

To turn off WebNFS service, enter the following command:

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

Displaying NFS Statistics

About Displaying NFS Statistics

The nfsstat Command Displays NFS and RPC Statistics

The `nfsstat` command displays statistics about NFS and Remote Procedure Calls (RPCs) 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.

Syntax

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

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

Options

Table 6-2 describes the options for the nfsstat command.

Table 6-2. nfsstat Command Options

Option	Description
none	When no options are specified, the command displays statistical information since the last time the filer was rebooted.
interval	When an interval is specified, the command displays statistics continually. The interval specifies the number of seconds the command waits between updates.

Table 6-2. nfsstat Command Options *(continued)*

Option	Description
`-h`	Displays statistics for a single client. You must provide the client's host name or IP address as an argument to the `-h` option. • To use the `-h` option, you must enable the `nfs.per_client_stats.enable` option by entering the following command: **`options nfs.per_client_stats.enable o`**n • Enable the per-client statistics collection mode 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.
`-l`	Displays statistics that have been collected for all NFS clients.
`-z`	Resets the statistics counters.

Example: no options

The following lines show the output of the `nfsstat` command when you specify no options:

```
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%
```

Example: using the -l option

The following lines show the output of the `nfsstat` command when you specify the `-l` option:

```
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%)
```

Example: using the -h option

The following lines show the output of the `nfsstat` command when you specify the `-h` option with a host name:

```
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%
```

Example: resetting counters with the -z option

The following command resets the counters:

```
nfsstat -z
```

CHAPTER 7
CIFS Administration

What Is CIFS?

CIFS (Common Internet File System) is a file-sharing protocol based on the Server Message Block (SMB) protocol widely in use by personal computers and workstations running a wide variety of operating systems. CIFS provides an open, cross-platform mechanism for client systems, including Windows systems, to request file and print services from server systems over a network.

This chapter describes how to manage the CIFS file protocol and users.

What You Can Do Only From the Filer Command Line or FilerView

You can do some operations that affect CIFS administration only from the filer command line or FilerView. These are

- Viewing volumes and examining their status. For additional information about volume operations, see Chapter 3, "Disk and File System Management."
- Setting or changing volume and qtree security style and oplocks status. For information about administering qtrees from the filer command line, see Chapter 12, "Qtree Administration." For information about how to administer qtrees in FilerView, see FilerView on-line help.

Effects of Renaming a Volume on Shares

If you change the name of a volume that contains at least one share, whether through Windows NT or the `vol rename` command, 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.

Scope of This Chapter

This chapter does not discuss procedures that take place on the clients and it does not describe how a machine joins a workgroup or domain. For information about these topics, refer to the manuals for your client operating systems or books about PC networking.

CIFS limitations

Introduction

This section describes CIFS limitations when operating on files on the filer.

User Manager Limitations

The Policy menu items and the New Users menu item are permanently disabled.

Server Manager Limitations

The following Server Manager features are not supported:

- stopping and starting services
- specifying the recipients of alerts

Limits on CIFS Open Files, Sessions, and Shares

Limits for the Dell PowerVault 720N, 740N, and 760N

The filer is subject to limits on file access through CIFS. Table 7-1 shows access limits for the 720N, 740N, and 760N filers with at least 120 MB of memory.

Table 7-1. CIFS File Access Limits

Type of access	Maximum number
Users	14,825
Files	269,500
Locked files	88,960
Shares	29,650

Changing or Viewing the Filer's Description

When to Change or View a Filer's Description

The description of a filer appears next to its name wherever a machine's description comments appear. Initially, the filer has no description. You might want to change the description to something more informative so that you can distinguish filers from each

other. You might want to view the description of a filer to find out, for example, what a particular filer does or who is in charge of it.

Changing a Filer's Description From Server Manager

To change the description of a filer from Server Manager, perform the following steps:

1. Open Server Manager by choosing it from the Start menu.
2. From the Server Manager Computer menu, choose Select Domain.

   Result: The Select Domain window opens.
3. In the Select Domain window, select the filer you want by typing its UNC name, for example, \\FILERNAME, in the Domain field, then clicking OK.
4. Double-click a filer name.

   Result: The Filer Properties window appears
5. Type a new description of the filer in the Description field.
6. Click OK.

   Result: The new description goes into effect.

Viewing a Filer's Description From the Filer Command Line

To view the description of a filer, enter the following command:

```
cifs comment
```

Changing a Filer's Description From the Filer Command Line

To change the description of a filer, enter the following command:

```
cifs comment "description"
```

where *description* is its description, which must be enclosed in quotation marks. The description must be no longer than 48 characters.

Adding CIFS Users to the Filer

When You Add CIFS Users

By default, Windows NT users who have an account map to a UNIX account of the same name. If they do not have a Windows NT account, the user becomes a generic user and uses the generic account, which is described in "Generic User Accounts." In

the rare case that you must add a user explicitly, you can do so with the methods described in this section.

You can add CIFS users to the filer at any time. The method you use for adding users to the filer depends on whether you are authenticating with a domain controller or the UNIX password database.

When Authenticating With a Domain Controller

To add a user, create an account for the CIFS user within your Windows NT domain environment. If you want the user to also use UNIX files, either create an entry in the */etc/passw*d file on the filer or include the user in the */etc/usermap.cfg* file.

What is the /etc/usermap.cfg file?

The */etc/usermap.cfg* file explicitly maps Windows NT users to the correct UNIX account and UNIX users to a Windows NT account. The file can be coded as follows:

- As a text file, with non-ASCII characters encoded in the NFS character set.
- As a UNICODE file created using the Windows NT tools Notepad or Microsoft Word.

The filer automatically detects which of these forms is in use.

Format of the /etc/usermap.cfg file

The format of the */etc/usermap.cfg* file is a list of text records in the following format:

```
[IP-qual:] [NT-domain\]NTUser [direction] [IP-qual:] UnixUser
```

Lines are processed sequentially.

Format variables

Table 7-2 describes the variables in the */etc/usermap.cfg* file description.

Table 7-2. */etc/usermap.cfg* **Format Variables**

Variable	Description
IP-qual	An IP qualifier that the filer uses in matching a user. You use an IP qualifier to narrow a match. *IP-qual* can be an IP address in any of the following formats: • A regular IP address specified as numbers separated by periods (dot notation) with an optional subnet address. For example, 192.4.1.0/24 narrows possible matches to the 192.4.1.0 class C subnet. • A host name. • A network name. • A network name with a subnet specified in dot notation. For example, corpnet/255.255.255.0 specifies the corpnet subnet.
NT-domain	Specifies the domain to match or the domain to use for a mapped UNIX account. The default is the domain in which the filer is installed.
NTUser	Any user-type account name. If the name contains a space, put the name in quotation marks.
direction	Restricts the direction of the mapping. By default, mappings are bidirectional. You can use one of three symbols: • => means NT to UNIX mapping only. • <= means UNIX to NT mapping only. • == means bidirectional mapping. Use this to explicitly indicate a bidirectional mapping.
UnixUser	A UNIX account name.

The following symbol conventions are in effect:

- An asterisk (*) matches any name.

  For example, to map all unmapped users to the UNIX "nobody" account, add the following line:

  ```
  *       nobody
  ```

- The null string ("") matches no name and rejects any user.

  For example, to prevent access completely, add the following line:

  ```
  *       ""
  ```

  This line prevents the default mapping of Windows NT users who have an account map to a UNIX account of the same name. Any lines after this line are disregarded by the filer.

- You can use either spaces or tabs as separators.

Name requirements

Windows NT and UNIX names have different requirements, as follows:

- Windows NT names are case-insensitive and can contain non-ASCII characters within the character set in the current code page. Windows NT user names can contain spaces, in which case you must enclose the name in quotation marks.
- UNIX user names are case-sensitive and must be in ASCII.

Default file contents

If the filer is domain authenticated, by default the */etc/usermap.cfg* file contains the following line:

```
domain\administrator    root
```

When Authenticating With the UNIX Password Database

To add a user, enter the user's information into the NIS password and group maps.

NOTE: If you do not use NIS, create entries for the user in the filer's /etc/passwd and /etc/group files.

Adding Local Groups to the Filer

How to Add a Local Group

You add a local group to the filer with the New Local Group window in the User Manager for Domains.

Adding a Group With the New Local Group Window

To create a new local group, perform the following steps:

1. Open User Manager for Domains by choosing it from the Start menu.
2. From the User menu, choose Select Domain.
3. Select the filer you want by typing its UNC name, for example, \\FILERNAME, in the Domain field, then clicking OK.

   Result: The User Manager window shows information for the specified filer.
4. From the New User menu, choose New Local Group.

   Result: The New Local Group window appears.
5. Type the name of the new group in the Group Name text box.
6. Type a description for the new group in the Description text box.

7. To add a member, type a user or group name in the Members list box or use the Add Users and Groups window, as described in Step 8.

Adding a name with the Add Users and Groups window

8. Click Add.

   Result: The Add Users and Groups window appears.

9. Click the arrow next to the List Names From text box to choose a domain that contains names that you want to add.

   Result: A list of names in the selected domain appears in the Names list box.

10. To add a name, type one or more user names in the Add Names list box or select one or more names and click Add.

11. Specify one or more names in either of these ways:

    - Click one or more names in the Names list box.
    - Type valid user or group names in the Add Names list box.

12. Click Add in the window where you specified the names.

    Result: The names are added to the Add Names field of the Add Users and Groups window.

13. If you want, display the full name of a user associated with an account name by clicking Show Full Name.

14. Click OK.

    Result: The names appear in the Members field of the New Local Group window.

Final steps in the New Local Group window

15. To remove one or more names from the list, select a name or names in the Names list, then click Remove.

16. Click OK to put the additions into effect.

Using CIFS Commands With a Remote Shell Program

What You Can Use a Remote Shell Program for

You can use a remote shell program, such as `rsh`, to

- execute CIFS commands
- create scripts containing CIFS commands to automate similar access rights tasks

UNIX Example

For example, using `rsh` on the administration host, you can set access rights for a specific user as follows:

```
rsh -l root -n filer cifs access home jsmith r-x
```

In this example, the filer name is filer, the user is jsmith, and the share is home. The user has read, execute, and browsing rights to the directory on the filer that has been defined as the home share.

Automating Access Rights

Because you can use CIFS commands through a remote shell program, 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 similar user information for each filer at your site.

Required Information in hosts.equiv File

Make sure that the following entries are added to the *hosts.equiv* file in the *etc* directory:

host *user*
host root

where *host* is the host you are using and *user* is your name.

NOTE: Make sure that you include both lines, or not all remote shell functionality works properly.

Enabling Guest and Generic Access

Two Ways to Give Access to Unauthenticated or Occasional Users

You can enable unauthenticated or occasional users to have access to the filer through CIFS in one of two ways:

- For users who are *not* in a trusted domain, use a guest account.
- For users who are authenticated but do not have an entry in the */etc/passwd* file on the filer, use a generic user account.

Guest Accounts

If you are using Windows NT domain 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.

Setting Up a Guest Account

To set up a guest account, use the following `options` command:

`options cifs.guest_account` ***`account_name`***

where *account_name* is the name of the guest account, usually *guest* or *nobody*, which is a preconfigured account in the */etc/passwd* file.

If you are using UNIX-style authentication, 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*.

Disabling Guest Access

To disable guest access, use the following `option` command:

`options cifs.guest_account`

`with a "" as` *`account_name`*`.`

Generic User Accounts

If you are using Windows NT domain authentication, a generic user account is mapped by default with the name "pcuser." The generic user account enables users who meet the criteria described in this section to connect to NTFS or mixed qtrees on the filer. For information about qtrees, see Chapter 10, "Qtree Administration," or FilerView on-line help.

NOTE: For generic user accounts to be active, pcuser must be an account in the /etc/passwd file.

Who Can Use the Generic User Account

To use the generic user account, a user must

- be authenticated
- be in a trusted domain
- not have an entry in the UNIX password database

All users of the generic user account have the same UNIX rights and the Windows NT rights granted by their Windows NT group membership because they appear as one account to the system.

Setting Up a Generic User Account

To set up a generic user account, use one of the following `options` commands with an account name to use as an argument as shown in Table 7-3:

Table 7-3. Generic User Account options Commands

User type	Option	Description
NT	`wafl.default_unix_user`	Specifies the UNIX user account to use when an NT user attempts to log in and that NT user would not otherwise be mapped.
UNIX	`wafl.default_nt_user`	Specifies the NT user account to use when a UNIX user accesses a file with NT security (has an ACL), and that UNIX user would not otherwise be mapped.

Disabling generic user access

To disable generic user access, use the `options` described in the previous table with a `""` as *account_name*.

Displaying a Filer's Shares

Ways to Share Folders

If you want to create a folder to be shared by CIFS clients, do one of the following:

- From Windows NT, use Server Manager as described in "Using Server Manager to Display a Filer's Shares."
- From the filer command line, use the `cifs shares` command to display share information, as described in "Using the cifs Shares Command to Display A Filer's Shares."

Using Server Manager to Display a Filer's Shares

To display a filers's shares with Server Manager and get detailed information, perform the following steps:

1. Open Server Manager by choosing it from the Start menu.
2. From the Server Manager Computer menu, choose Select Domain.

   Result: The Select Domain window opens.

3. In the Select Domain window, select the filer you want by typing its UNC name, for example, \\FILERNAME, in the Domain field, then clicking OK.

   Result: A Server Manager window for the filer appears.

4. Choose Computer > Shared Directories.

   Result: The Shared Directories window appears.

5. To get additional details about a share, double-click a share name.

   Result: The Share Properties window appears.

Using the cifs shares Command to Display a Filer's Shares

To display the filer's list of shares from the filer's command line, use the `cifs shares` command.

Command Syntax

The syntax is as follows:

```
cifs shares
```

Example of Displaying a filer's Shares

The following example shows a filer's shares:

```
cifs shares
Name          Mount Point                      Description
----          -----------                      -----------
HOME          /vol/muffin/home                 Default Share
                        everyone / Full Control
                        techpubs{g} / Full Control
C$            /vol/muffin                      Remote
Administration
                        everyone / Full Control
                        BUILTIN\Administrators / Full Control
openhome      /vol/muffin/writers_home         readable home
dirs
              ... user limit=1200
              ... forcegroup=techpubs
                        everyone / --x
                        techpubs{g} / r-x
stock         /vol/muffin                      Not half-baked
              ... user limit=10
                        everyone / Full Control
flour         /vol/muffin/blueberry            Flour power
                        everyone / Full Control
sesame        /vol/bagels/
                        everyone / Full Control
```

Creating and Changing a Share

Ways to Share Folders

If you want to create a folder to be shared by CIFS clients, do one of the following:

- From Windows NT, use Server Manager as described in "Creating a Share From Server Manager."
- From the filer command line, use the `cifs shares` command to define a new share, as described in "Creating a Share with the cifs Shares Command."

NOTE: By default, three shares are created during CIFS setup: C$, IPC$, and HOME. In filer console displays, the C$ share corresponds in UNIX to /vol/vol0 and the HOME share corresponds to /vol/vol0/home.

Creating a Share From Server Manager

To create a share from the Windows NT desktop using Server Manager, create a folder, then share it by performing the following steps:

1. Create a folder on the filer.
2. Open Server Manager by choosing it from the Start menu.
3. From the Server Manager Computer menu, choose Select Domain.

   Result: The Select Domain window opens.
4. In the Select Domain window, select the filer you want by typing its UNC name, for example, \\FILERNAME, in the Domain field, then clicking OK.

   Result: A Server Manager window for the filer appears.
5. Choose Computer > Shared Directories, then click the New Share button.

   Result: The New Share window appears.
6. Type a name for the new share in the Share Name field. The share name is case-sensitive.
7. Type the local path here of the folder you created in Step 1. This is usually C:*VOL**VOLNAME**FOLDER*.
8. Type a description of the share in the Comments field, if you want.
9. If you do not want to limit the number of users that can connect to the share at the same time, select Maximum Allowed under User Limit.
10. To limit the number of users that can connect to the share at the same time, click the arrows next to Users until the desired number appears.
11. If the share has Windows NT security, click Permissions to set permissions.

12. Click OK.

    Result: The New Share window disappears and the share is created.

Changing the Share Description and User Limit With Server Manager

To change the description and user limit with Server Manager, perform the following steps:

1. Open Server Manager by choosing it from the Start menu.
2. From the Server Manager Computer menu, choose Select Domain.

   Result: The Select Domain window opens.
3. In the Select Domain window, select the filer you want by typing its UNC name, for example, \\FILERNAME, in the Domain field, then clicking OK.

   Result: A Server Manager window for the filer appears.
4. Choose Computer > Shared Directories.

   Result: The Shared Directories window appears.
5. Double-click a share name.

   Result: The Share Properties window appears.
6. Type a new description of the share in the Comments field, if you want.
7. If you do not want to limit the number of users that can connect to the share at the same time, select Maximum Allowed under User Limit.
8. To limit the number of users that can connect to the share at the same time, click the arrows next to Users until the desired number appears.
9. Click Permissions to change share-level permissions.
10. Click OK.

    Result: The window disappears and the new values go into effect.

Creating a Share With the cifs shares Command

Following is the syntax for the `cifs shares` command for creating a share:

```
cifs shares -add sharename path [ -comment description ]
[ -forcegroup groupname ] [ -maxusers n ]
```

Table 7-4 describes the parameters.

Table 7-4. Creating a Share With cifs shares Command

Variable	Description
description	A string describing the purpose of the share. It must contain only characters in the current code page. 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, you must enclose it in single quotation marks.
groupname	The name of the group you want all files in the share to get the group membership of.
n	The maximum number of users that you specify can connect to the share at the same time. The limit on this number is dependent on filer memory, and is shown in "Limits on CIFS Open Files, Sessions, and Shares."
path	The path name, relative to the root of the filer's file system, of the root directory of the share. *NOTE: 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 (/).*
sharename	The name of the share, which is used by CIFS users to obtain access to the directory on the filer. If *sharename* already exists, the `cifs shares -add` command fails. **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.**

Example

The following example creates a new share called *library*:

```
cifs shares -add library /vol/vol0/home/lib 'New file library'
```

This example creates the library share and defines it as HOME\LIB (/home/lib in UNIX notation) in the filer's root volume. With the appropriate access rights, CIFS users can gain access to the HOME\LIB directory in the root volume, which is displayed as the library share on their computers. For more information about setting access rights, refer to "Assigning and Changing Access Rights."

Using the cifs shares Command to Change the Share

To change the description, forced file ownership, and user limits of a share, use the following command:

```
cifs shares -change sharename [ -comment description | -nocom-
ment ] [ -forcegroup groupname | -noforcegroup ]
[ -maxusers n | -nomaxusers ]
```

Table 7-5 describes the parameters.

Table 7-5. Changing a Share With cifs shares Command

Parameter or variable	Description
description	A string describing the purpose of the share. It must contain only characters in the current code page. It is displayed in the share list in Network Neighborhood on the client. If the description contains spaces, you must enclose it in single quotation marks.
groupname	The name of the group you want all files in the share to get the group membership of.
n	The maximum number of users that you specify can connect to the share at the same time. The limit on this number is dependent on filer memory, and is shown in "Limits on CIFS Open Files, Sessions, and Shares."
-nocomment	Specifies no description.
-noforce-group	Specifies no particular group to own the files that are created in the share.
-nomaxusers	Specifies no maximum number of users that can connect to the share at the same time.
sharename	The name of the share, which is used by CIFS users to obtain access to the directory on the filer. It must contain only characters in the current code page.

Displaying Information About Shares

Methods of Displaying Information About Shares

You can display information about a specific share or all shares on the filer.

To display information about shares, do either of the following:

- From the Windows NT desktop, use Server Manager as described in "Using Server Manager to View Information about Shares."
- From the filer, use the cifs shares command as described in "Using the cifs Shares Command to View Information about Shares."

Using Server Manager to View Information About Shares

To display information about shares from Server Manager, perform the following steps:

1. Open Server Manager by choosing it from the Start menu.
2. From the Server Manager Computer menu, choose Select Domain.

   Result: The Select Domain window opens.
3. In the Select Domain window, select the filer you want by typing its UNC name, for example, \\FILERNAME, in the Domain field, then clicking OK.

   Result: A Server Manager window for the filer appears.
4. Double-click the name of a filer.

   Result: The Filer Properties window appears.
5. Click the Shares button.

   Result: Information about each share on the filer appears.

Using the cifs Shares Command to View Information About Shares

To display information about a share or shares that exist, enter the following command:

```
cifs shares sharename
```

where *sharename* is the specific name of the share about which you want information. If you want information about all shares, leave *sharename* blank.

Examples of Displaying Share Information

The following command example displays the share information only for the share library created by the cifs shares -add command described in "Creating a Share with the cifs Shares Command."

```
cifs shares library
```

The following command displays information about all shares:

```
cifs shares
```

The sample output is from `cifs shares`:

```
Name          Mount Point                       Description
----          -----------                       -----------
HOME          /vol/muffin/home                  Default Share
                        everyone / Full Control
                        techpubs{g} / Full Control
C$            /vol/muffin                       Remote
Administration
                        everyone / Full Control
                        BUILTIN\Administrators / Full Control
openhome      /vol/muffin/writers_home          readable home
dirs
              ... user limit=1200
              ... forcegroup=techpubs
                        everyone / --x
                        techpubs{g} / r-x
stock         /vol/muffin                       Not half-baked
              ... user limit=10
                        everyone / Full Control
flour         /vol/muffin/blueberry             Flour power
                        everyone / Full Control
sesame        /vol/bagels/
                        everyone / Full Control
```

Deleting a Share

How to Delete a Share

To delete shares, do either of the following:

- From the Windows NT desktop, use Server Manager as described "Using Server Manager to Delete a Share."
- From the filer, use the cifs shares command, as described in "Using the cifs Shares Command to Delete Shares."

Using Server Manager to Delete a Share

To delete a share from Server Manager, follow these steps:

1. Open Server Manager by choosing it from the Start menu.
2. From the Server Manager Computer menu, choose Select Domain.

   Result: The Select Domain window opens.
3. In the Select Domain window, select the filer you want by typing its UNC name, for example, \\FILERNAME, in the Domain field, then clicking OK.

   Result: A Server Manager window for the filer appears.

4. In the Server Manager window for the filer, choose Computer > Shared Directories, then click the Properties button.

   Result: The Share Properties window appears.

5. Click Stop Sharing, then click OK.

   Result: The folder is no longer shared.

Using the cifs shares Command to Delete Shares

To delete a share, use the `cifs shares -delete` command.

Command Syntax

The `cifs shares -delete` command has the following syntax:

`cifs shares -delete` *`sharename`*

where *sharename* is the specific name of the share that you want to delete.

Example

The following command deletes the share created by the cifs shares -add command described in "Creating a Share with the cifs Shares Command."

```
cifs shares -delete library
```

Creating a Home Share for Each User

When to Create a Home Directory

You can create a share that contains home directories of registered CIFS users.

For example, if there are users called *user1* and *user2*, the share contains directories for user1 and user2. When user1 connects to the filer and asks for its list of shares, the display shows a share called user1, but not user2 or any other individual user.

NOTE: If Domain1\user and Domain2\user are the same, they do not have different home directories. To prevent access by the wrong user, set Windows permissions and UNIX permissions at the root of the user's home directory.

Accessing a Home Directory

Users access their home directories in the same way as any other share. That is, users can open the share with Network Neighborhood, by mapping a drive, or by using a UNC name. The UNC name is

`\\filer\`*`username`*

Share Name Length Limitations

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 *administrator*.

If there were users *administrator* and *administrator1*, they would see an offer for their own home directory, but both would see it as an offer to share the name *administrato*. Therefore, each user 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*

   Where *homedirpath* is the UNIX path name that will be mapped to the share CIFS.HOMEDIR. This share automatically contains CIFS user home directories.

   If the directory containing the home directories is */vol/vol1/homes*, enter the following command:

   options cifs.home_dir /vol/vol1/homes

2. Within the directory specified by the `cifs.home_dir` option, create a directory for each user. Make sure that the following conditions are met:

   - Each directory name matches the user's login name exactly.
   - Each user is the owner of the directory.

Creating Share Home Directories

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, follow this procedure:

1. Create an additional share, from either Windows or the filer, that maps to the same path name as the CIFS.HOMEDIR share.
2. From either Windows or the filer, assign each user the appropriate access permissions to other users' home directories.

Example From the Filer

The following example shows how to create an additional share, assign user access rights to the share, and display the share information from the filer command line. If authentication is through the */etc/passwd* file, UNIX permissions are shown; otherwise, Windows NT permissions appear.

```
cifs shares -add enghomes /vol/vol1/homes \
   -comment "Readable home directories"

"Readable home directories"

cifs access enghomes -g engineering r-x

cifs shares
Name         Mount Point                      Description
----         -----------                      -----------
enghome      /vol/vol1/homes                  Readable home
directories
                        engineering{g} / r-x
C$           /vol/vol0                        Remote
Administration
                        BUILTIN\Administrators / Full Control
HOME         /vol/vol0/home                   Default Share
                        everyone / Full Control
CIFS.HOMEDIR /vol/vol1/homes                  Home Directories
                        everyone / Full Control
```

Result

Users in the *engineering* group can read all home directories in *HOMES* on the root volume, 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.

Assigning and Changing Access Rights

When to Assign or Change Access Rights

After you create a share, you define the user or group access rights to the share. If a group or a user no longer exists, you can remove the corresponding entry from an ACL.

Methods of Assigning or Changing Access Rights to a Share

To assign access rights, use either of the following methods:

- From Windows, follow the instructions in "Assigning or Changing Access Rights with Server Manager."

- From the filer command line, do one of the following:

  a. Add access with the cifs access command, as described in "Giving Access With the Cifs Access Command."

  b. Remove a user or group with the cifs access -delete command, as described in "Removing a User or Group With the cifs Access -delete Command."

Assigning or Changing Access Rights With Server Manager

To assign or change access rights with Server Manager, use the Access Through Share Permissions window by following these steps:

1. Open Server Manager by choosing it from the Start menu.
2. From the Server Manager Computer menu, choose Select Domain.

   Result: The Select Domain window opens.
3. In the Select Domain window, select the filer you want by typing its UNC name, for example, \\FILERNAME, in the Domain field, then clicking OK.

   Result: A Server Manager window for the filer appears.
4. In the Server Manager window, click Computer > Shared Directories, then click the Properties button.

   Result: The Share Properties window appears.
5. Click Permissions.

   Result: The Access Through Share Permissions window appears.
6. To add a new user or group, click Add.

   Result: The Add Users and Groups window appears.

Adding a name with the Add Users and Groups window

7. Click the arrow next to the List Names From field to choose a domain that contains names that you want to add.

   Result: A list of names appears in the Names list box.
8. Specify one or more names in either of the following two ways:

   a. Click one or more names in the Names list box.

   b. Type valid user or group names in the Add Names text box.
9. Click Add in the window where you selected the names.

   Result: The names are added to the Add Names field of the Add Users and Groups window.

10. If you want, add or modify the access type by selecting one or more names and choosing an access type from the Type of Access list.

Steps in the Access Through Share Permissions window

11. To assign or change an access type, select a name or names in the Names list, then click the arrow next to Type of Access and select an access type.

12. To remove one or more names from the list, select a name or names in the Names list, then click Remove.

13. Click OK.

Final step in the Properties window

14. Click OK to put the changes into effect.

Giving Access With the cifs access Command

You use the `cifs access` command to assign access to a share from the filer command line. To change the permissions, run the command with the new permissions.

Command syntax

The `cifs access` command has the following syntax:

```
cifs access share [ -g ] user|group rights
```

The `-g` flag specifies that the access rights are defined for a group.

rights can be UNIX-style permissions or Windows NT-style rights:

- UNIX-style permissions are defined as `r`, `w`, and `x`, which mean read, write and delete, and execute and browse, respectively. To deny a right to a user, use a hyphen (-).
- Windows NT-style rights are No Access, Read, Change, and Full Control.

Examples

Here are some examples of assigning access rights from the filer:

- `cifs access library -g engineering rwx`
- `cifs access library domain\joed Change`

NOTE: The group everyone is reserved. When you use it 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, you enter the following command:
`cifs access library everyone rwx`

Removing a User or Group With the cifs access -delete Command

If a user or group no longer exists, you can remove the corresponding entry from an ACL. Use the following command syntax to remove an entry in an ACL:

```
cifs access -delete share user | group
```

Following are some examples of removing entries from the *library* share.

The following command removes the *engineering* group from the *library* share.

```
cifs access -delete library engineering
```

The following command removes the user *joed* from the *library* share.

```
cifs access -delete library domain\joed
```

Displaying Access Rights to an NTFS File

Access Rights Display Methods

You can display access rights to an NTFS file from Windows.

Displaying Access Rights From the Windows Desktop

To display access rights to a file from the Windows desktop, follow these steps:

1. Right-click a file and choose Properties from the pop-up menu.

   Result: The Properties sheet appears.

2. Click the Security tab.

   Result: If the file is a Windows file, the Security sheet appears.

3. Click Permissions.

   Result: Permissions are displayed.

Changing UNIX Permissions and DOS Attributes From Windows

How to change UNIX permissions

Some files and directories on the filer have both Windows and UNIX-style permissions. You can change Windows-style permissions from Windows by editing Windows permissions. To change UNIX permissions in a UNIX file system from the Windows desktop or to change DOS attributes, you use the SecureShare Access tool.

To use the SecureShare Access tool from a client, you must install the tool on the client. For instructions about downloading the SecureShare Access tool, see the *Software and Firmware Upgrade Guide* or *Start Here*.

NOTE: To change UNIX permissions, you must understand them. Explaining UNIX permissions is outside the scope of this guide. Consult literature about UNIX for an explanation of UNIX permissions.

Displaying SecureShare Access

To display SecureShare Access, follow these steps.

1. Select the files and directories whose permissions you want to change.

   The items are highlighted.

2. Right-click one of the items you selected.

   A pop-up menu appears.

3. Choose Properties from the pop-up menu.

   The Properties dialog box appears.

4. Click the SecureShare tab.

   SecureShare Access appears.

Changing the Permissions of a Single Item

If you select only one item, SecureShare Access appears.

CAUTION: SecureShare Access has no undo feature. Use it very carefully.

qtree Security Style Effects

The security style of the qtree from which you select an item has the following effects:

- In NTFS-style qtrees, all the edit fields and check boxes are disabled. Use standard Windows NT tools to manipulate NTFS files.
- In NFS-style qtrees, the Has ACL check box is meaningless and is disabled.
- In mixed-style qtrees, if a file has an ACL, the ACL is removed when you click the OK or Apply button. A dialog box prompts you to confirm that this is what you want.

NOTE: If you click OK or Apply on any page or tab of a Properties sheet, it puts into effect all changes that you made on all pages or tabs of the Properties sheet.

Recursive Application of Changes

If one or more of the files is a directory, the Descend Into Subdirectories check box is enabled. Select this check box to apply the changes you make recursively to the contents of any directory you selected. Links are not changed, but their targets are.

Changing the Permissions of Multiple Items

You can change the permissions of several items at once by selecting them, then displaying SecureShare Access by following Steps 2 through 4 in "Displaying SecureShare Access."

In the multiple-item display, the following conditions are in effect that are different than the single-item display.

- Permissions; DOS attributes; or characteristics; for example, whether an item is a directory, that are common to all the items you selected appear with a white background.
- Permissions; DOS attributes; or characteristics; for example, whether an item is a directory, where one item differs from the rest of the items you selected appear with a light gray background.
- Permissions or DOS attributes that you set apply to all items you selected.
- Turning one of the check boxes gray preserves the permissions on any of the selected items.

Sending a Message to All Users on a Filer

When to Send a Message

You might want to send a message to all users on a filer to tell them of important events. The message appears in an alert box. For example, you might need all users to close any open files on the filer, but not want to stop CIFS services and, therefore, not take advantage of the messaging function of the `cifs terminate` command.

How to Send the Message

To send a message to all users on a filer, follow these steps:

1. Open Server Manager by choosing it from the Start menu.
2. From the Server Manager Computer menu, choose Select Domain.

   Result: The Select Domain window opens.
3. In the Select Domain window, select the filer you want by typing its UNC name, for example, \\FILERNAME, in the Domain field, then clicking OK.

   Result: A Server Manager window for the filer appears.

4. In the Server Manager window, from the Computer menu, choose Send Message.

   Result: The Send Message window appears.

5. Enter a message in the text box of the Send Message window.

6. Click OK to send the message.

Event Auditing

You Can Audit File Access Events

Data ONTAP 5.3 enables you to monitor reads and writes of a specified file on the filer by a specified user. The procedure for doing so is the same as for Windows NT.

NOTE: The file on the filer must be in a mixed or ntfs volume or qtree. You cannot audit events on a file in a UNIX volume or qtree.

You can specify the logging of successes, failures, or both for each type of event as described in "Setting a System ACL on a File." You can view the events on the filer using the Windows Event Viewer, as described in "Viewing Events in a Security Log."

Why Use Event Auditing

You use event auditing to troubleshoot access problems, to check for suspicious activity on a system, or to investigate a security breach.

- If someone who should have access to a file cannot get access, examining the event can provide a clue to resolving the problem.
- If a file has been changed in a way that indicates a security breach, event auditing might provide clues about the nature of the breach if subsequent changes occur.

Active Event Log Naming

The active event log is the file to which system writes access logging information. The default active event log is */etc/log/adtlog.evt*. You can specify another active event log with the `cifs.access_logging.filename` option. You can change active event logs at any time. The other logs remain available for reading until you delete them.

Log Access

Anyone can read an event log. You cannot write to an event log or clear one, but you can delete one.

Event Log Detail Displays

How to Examine an Event in Detail

You examine an event in detail by double-clicking it and looking at the resulting event details display. There are the following kinds of event details displays:

- Windows file access detail displays
- UNIX file access detail displays
- Unsuccessful file access detail display
- Lost record event detail display

Windows File Access Detail Displays

Windows file access detail displays show the following information displayed in the detail screens as shown in Table 7-6

Table 7-6. Windows File Access Detail Displays

Field	Description
Object Server	The name of the subsystem server process calling the audit check function. This is always SECURITY, because this is a security log.
Object Type	The type of object being accessed.
Object name	The name (such as a file name) of the object being accessed.
New Handle ID	The new handle identifier of the open object.
Operation ID	A unique identifier associating multiple events resulting from a single operation.
Process ID	The identifier of the client process accessing the object.
Primary User Name	The user name of the user requesting the object access. When impersonation is taking place, this is the user name with which the server process is logged on.
Primary Domain	The name of the computer, or SYSTEM if the user identified by Primary User Name is SYSTEM. If the computer is a member of a Windows NT Server domain, this can also be the name of the domain containing the primary user's account.
Primary Logon ID	A unique identifier assigned when the primary user logged on.

Table 7-6. Windows File Access Detail Displays *(continued)*

Field	Description
Object User Name	Your login.
Client Domain	The name of your computer or the domain containing the client user's account.
Client Logon ID	A unique identifier assigned when the client user logged on.
Accesses	The types of accesses to the object that were attempted.
Privileges	Your privileges.

UNIX File Access Detail Displays

UNIX file access detail displays show the same kind of information as the Windows file access event detail screens, but instead of an object name, NFS access appears because the file is being accessed through NFS. In addition, UNIX file access detail displays show the following information about the file that you are monitoring:

- The ID of the volume in which the file is located
- The ID of the latest snapshot in which the file is located
- The inode of the file

This information enables you to find the file using the `ls -i` command.

Unsuccessful File Access Detail Display

An unsuccessful file access detail display appears when a user could not access a file. For example, an unsuccessful file access occurs when a user tries to access a file but does not have permission to access it.

The display shows the ID of the user who tried to access the file and an indication that the access attempt was unsuccessful.

Lost Record Event Detail Display

If the system could not create an audit record, the Lost record event detail display gives a reason, such as the following reason:

```
Internal resources allocated for the queueing of audit messages
have been exhausted, leading to the loss of some audits.
      Number of audit records discarded:          1
```

Event Auditing Overview

Description

This overview of how to audit events contains steps that are described in detail in later sections.

Steps

To audit events, complete the following steps:

1. Enable CIFS access logging, as described in "Enabling CIFS Access Logging."
2. Set a system ACL on the files you want to audit, as described in "Setting a System ACL on a File." This involves specifying the users or groups whose access to the file you want to audit.
3. Use the Event Viewer, as described in "Viewing Events in a Security Log." to view the security log.

Enabling CIFS Access Logging

Description

Use this procedure to enable CIFS access logging. You follow this procedure when you want to start logging file access events by a specified user on a particular file. After you complete this procedure, the filer is ready to process access logging information.

Step

To enable CIFS access logging, enter the following command:

```
options cifs.access_logging.enable on
```

Disabling CIFS Access Logging

Description

Use this procedure to disable CIFS access logging. You follow this procedure when you want to stop logging file access events and free the resources that are used by access logging. After you complete this procedure, access logging stops.

Step

To disable CIFS access logging, enter the following command:

```
options cifs.access_logging.enable off
```

Specifying the Active Event Log

Description

Use this procedure to specify the active event log. You follow this procedure when you want to specify the name of a file to which the system writes access logging information. You can do this at any time. After you complete this procedure, if access logging is enabled, the system writes access logging information to the file you specified.

Prerequisites

You must have the path name of the file to use. This file must be in an existing writable directory.

Step

To specify the active log, center the following command at the command line:

`options cifs.access_logging.filename` ***`path_name`***

path_name is the path name in UNIX format of the file to which you want the system to write access logging information.

Setting a System ACL on a File

Description

Use this procedure to set a system ACL (SACL) on a file. You must complete this procedure to monitor access activity on a file. You complete this procedure when you have decided to monitor access to a file by specified users or groups. The procedure is the standard Windows NT procedure for setting a SACL on a file. After you complete this procedure, if access logging is enabled, the filer logs accesses to the file by the users or groups you specified.

Prerequisites

You must have the following items to complete the procedure:

- The name of the file you want to monitor
- The name of the users or groups whose access to the file you want to monitor

Steps

To set a SACL on a file, complete these steps;

1. Right-click the file you want to monitor.

   Result: The Properties tab for that file appears.

2. Click the Security tab.

   Result: The Security properties window appears.

3. Click the Auditing button.

   Result: A blank File Auditing window appears for the file you specified.

4. Click the Add button.

   Result: The Add Users and Groups window appears.

5. Click the arrow next to the List Names From text box to choose a domain that contains names that you want to add.

   Result: A list of names in the selected domain appears in the Names list box.

6. To add a name, type one or more user names in the Add Names list box or select one or more names and click Add.

7. Specify one or more names in one of these ways:

   - Click one or more names in the Names list box.
   - Type valid user or group names in the Add Names list box.

8. Click Add in the window where you selected the names.

   Result: The names are added to the Add Names field of the Add Users and Groups window.

9. If you want, display the full name of a user associated with an account name by clicking Show Full Name.

10. Click OK.

    Result: The names appear in the Name field of the File Auditing window and the Events to Audit check boxes are enabled.

NOTE: Execute, Delete, Change Permissions, and Take Ownership events are not currently supported.

Viewing Events in a Security Log

Description

Use this procedure to view events in a security log and check the events on the file and users you specified in the procedure "Setting a System ACL on a File." After you complete this procedure, access information is displayed.

Prerequisites

You should have the name of the security log that you want to view.

Steps

To view events in a security log, perform the following steps:

1. Open the Event Viewer by selecting the following menu items: Programs, Administrative Tools, and then Event Viewer.

   Result: The event viewer appears.

2. From the Log menu of the Event Viewer, choose Open.

   Result: An Open window appears.

3. Choose the share that contains the log you want to look at.

   Result: A list of files appears.

4. Click the log file you want to examine. An Open File Type window appears.

5. If Security is not selected, select it.

6. Click OK.

   Result: The Event viewer appears and displays a list of Security events.

   In the display, symbols preceding event entries have the following meanings:

   - Key - Successful access attempts
   - Lock - Unsuccessful access attempts

7. To view an event detail, double-click the event.

   Result: An event detail screen appears.

Using Oplocks

What Oplocks Do

Oplocks (opportunistic locks) enable the redirector on a CIFS client in certain file-sharing scenarios to perform client-side caching of read-ahead, write-behind, and lock information. A client can then work with a file (read or write it) without regularly reminding the server that it needs access to the file in question. This improves performance by reducing network traffic.

When to Use Oplocks

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.

Data Loss Possibilities

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.

Error Handling And Write Completion

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.

When to Turn Oplocks Off

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 even the slightest data loss.

Otherwise, you can leave CIFS oplocks On.

Turning Oplocks On and Off Globally

You can turn CIFS oplocks On or Off globally for the entire filer or for individual qtrees, which are special directories that are described in detail in Chapter 10, "Qtree Administration."

Turning Oplocks Off

You turn all CIFS oplocks Off with the following `options` command:

```
options cifs.oplocks.enable off
```

Turning Oplocks On

You turn CIFS oplocks On with the following `options` command:

```
options cifs.oplocks.enable on
```

Turning Oplocks On or Off at Individual Clients

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.

For Additional Information

For additional information about oplocks, consult the Microsoft Knowledge Base at *http://www.microsoft.com/kb*.

Displaying CIFS Statistics

How and Why to Display CIFS Statistics

You use the `cifs stat` command to display statistics about CIFS operations that take place on your filer. You use the `cifs stat` display for diagnostic purposes.

Statistics Displays With the cifs stat Command

You can use the `cifs stat` command in two forms:

- If you specify a time interval, the command displays statistics at the specified intervals.
- If you do not specify a time interval, the command displays CIFS statistics that have accumulated since the last reboot.

Example Of cifs stat Output

The following command displays statistics every second:

```
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
```

Table 7-7 describes the fields.

Table 7-7. cifs stat Command Output Fields

Field	Description
`GetAttr`	Attribute operations
`Read`	Read data operations
`Write`	Write data operations
`Lock`	Lock operations
`Open/Cl`	Open and close operations
`Direct`	Directory operations
`Other`	Other operations, such as deletes and logoffs

Displaying CIFS Session Information

CIFS Session Information You Can Display

You can display information about connected users and the number of shares and open files for each user. You can also display information about a specific connected user.

Displaying Information With the cifs sessions Command

The `cifs sessions` command syntax is as follows:

```
cifs sessions <username>
```

Displaying Information About All Connected Users

To display information about all connected users, use the following command syntax:

```
cifs sessions
```

Sample output is

```
Server Registers as 'SILVER' in group WNT-DOMAIN
WINS Server: 272.320.0.4
PC style Access Control is being used
Using domain controller WNT-DOMAIN-PDC for authentication
=======================================
PC (user)           #shares       #files
SMITHPC (qsmith)      1             1
PETERSPC (zpeters)    2             3
```

Displaying Information About One User

To display the information for a particular user, specify the user name in the command; for example:

```
cifs sessions ghopper

users
    shares/files opened

  HALEY-HOME1   (ghopper)
    ENG-USERS
              Read-denyW   -
\GHOPPER\SRC\PROD\COMMON\HTTPD\httpd_fast.c


  HALEY-PC      (ghopper)
    ENG-USERS
```

Displaying Connected User Security Information

To display security information for each connected user, use `cifs sessions` with the `-s` option. After the first two lines, detailed information for each connected user is displayed. The following example lists only one user.

```
cifs sessions -s

users
  Security Information

HOLARD-PC     (root)
        ***************
        UNIX uid = 0
        user is a member of group daemon (1)
        user is a member of group www (204)
        user is a member of group well (0)
        user is a member of group http (500)
```

```
NT membership
        WNT-DOMAIN\root
        WNT-DOMAIN\Domain Users
        WNT-DOMAIN\Domain Admins
        WNT-DOMAIN\SU Users
        WNT-DOMAIN\Installers
        BUILTIN\Users
        BUILTIN\Administrators
User is also a member of Everyone, Network Users,
Authenticated Users
***************
```

Stopping and Restarting CIFS Sessions

Ways to Stop CIFS Sessions

If you want to stop CIFS sessions for all clients or for a single client, do one of the following:

- From Windows NT, use Server Manager as described in "Disconnecting Users With Server Manager."
- From the filer command line, use the cifs shares command to display share information, as described in "Using the cifs Terminate Command."

Disconnecting Users With Server Manager

To stop CIFS sessions with Server Manager, follow these steps:

1. Open Server Manager by choosing it from the Start menu.
2. From the Server Manager Computer menu, choose Select Domain.

   Result: The Select Domain window opens.
3. In the Select Domain window, select the filer you want by typing its UNC name, for example, \\FILERNAME, in the Domain field, then clicking OK.

   Result: A Server Manager window for the filer appears.
4. In the Server Manager window, double-click the name of a filer.

   Result: The Properties window for the filer appears.
5. Click Users.

   Result: The User Sessions window appears.

6. To disconnect one or more users, do one of the following:

   - To disconnect a single user or selected users, select them, then click Disconnect.
   - To disconnect all users, click Disconnect All.

   Result: The selected users are disconnected.

NOTE: If at least one of the selected users has open resources, an alert box appears for you to confirm or cancel your command.

Using the cifs terminate Command

You can stop CIFS service for a specific client or shut down CIFS service from the filer by using the `cifs terminate` command. Always terminate all CIFS sessions before you reboot or turn Off the filer.

You can specify a single client or all clients, and the time delay, in minutes, before the CIFS sessions are terminated, as shown in the following command syntax:

```
cifs terminate [client] [[-t] time]
```

Table 7-8 describes the variables.

Table 7-8. cifs terminate Command Variables

Variable	Description
client	Name of the client for which you are ending a CIFS session.
time	Number of minutes before the termination happens.

The cifs terminate Command Not Persistent

The `cifs terminate` command disables CIFS sessions 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 the */etc/cifsconfig.cfg* file from the filer or rename the file.

Time Delay

You can delay the termination of CIFS service after you enter the `cifs terminate` command.

Default Time Delay

The default time delay is five minutes.

Changing the Time Delay

When you use the `-t` option, the command counts down from the time specified.

Canceling the cifs terminate Command

If you want to cancel the `cifs terminate` command, press Ctrl-C before the end of the countdown.

NOTE: The `halt` *command automatically invokes the* `cifs terminate` *command.*

CAUTION: The `reboot` **command also stops CIFS service; however, it does not provide a time delay during which users can save their open files before the disconnect. 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.**

Examples of the cifs terminate Command

Here are some examples of the `cifs terminate` command.

Terminating CIFS Service for All Users on the Filer

To terminate CIFS service for all users on the filer after 10 minutes, enter the following command:

```
cifs terminate -t 10
```

The `cifs terminate` command displays an alert message on CIFS clients that warns them of the pending shutdown of CIFS service.

NOTE: Windows 9x and Windows for Workgroup clients must have the WinPopup program configured before they can display the alert message.

Console Display

Here is the display for `cifs terminate` when you do not specify a time:

```
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
```

Terminating a CIFS Session for a Specific Client

To terminate a CIFS session for a particular client, specify the name of the computer in the command. For example, the following command terminates a CIFS session for a computer named PETERSPC after 10 minutes:

```
cifs terminate PETERSPC -t 10
```

Using the cifs restart Command to Restart CIFS Service

To restart CIFS service, use the `cifs restart` command, as follows:

```
cifs restart

CIFS server is registering...
CIFS server is running.
```

Reconfiguring the Filer for CIFS

When to Reconfigure a Filer for CIFS

You can reconfigure the filer for CIFS service at any time, for example, if you want to change the authentication method from PDC (Primary Domain Controller) to UNIX password database.

How to Reconfigure a Filer for CIFS

To reconfigure a filer for CIFS, follow these steps:

1. Enter the `cifs terminate` command to stop CIFS service.
2. Enter the `cifs setup` command to reconfigure CIFS service.
   - If you enter only valid information, when you exit the program, the filer automatically restarts CIFS using the new configuration information.
   - If you enter some invalid information, for example, you mistype a domain name, when you exit the program, the filer restarts CIFS with the previous configuration.

CHAPTER 8
HTTP Administration

NOTES: You can use the filer as an HTTP server only if you 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.

As with UNIX-based systems, the URL is case sensitive.

Starting HTTP Service

Procedure for 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 */etc/log/httpd.log* log file 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" in Chapter 7 to configure your filer so that the appropriate MIME Content-Type header is sent with each file transferred.

Procedure for Testing HTTP Service

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 *filer* and the root directory for HTTP is */vol/vol0/home/users/pages*, enter this URL:

   `http://filer/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://filer/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

Configuration Files for Password Protection

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

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

Specifies a "realm," that is, an alias for the directory that 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

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

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 the *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 the user bob:

```
<Directory /vol/vol0/home/htdocs/private/spec>
AuthName polard Private Stuff
<Limit GET>
require user bob
</Limit GET>
</Directory>
```

The <Limit GET> and </Limit GET> directives, which might have been imported from an Apache or NCA Web server, are not supported. 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
- rose

Using the HTTP Virtual Firewall

About the HTTP Virtual Firewall

The HTTP virtual firewall feature enables you to maintain security on your filer.

You can 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.

Syntax

Mark an interface as untrusted or trusted by setting an option to the `ifconfig` command. Following are examples of the command:

- To mark the *e0* interface as untrusted, enter the following command:

  ```
  ifconfig e0 untrusted
  ```
- To mark the *e0* interface as trusted, enter the following command:

  ```
  ifconfig e0 trusted
  ```

Using Virtual Hosting

About 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 Set Up and Enable Virtual Hosting

To enable virtual hosting, you

- direct HTTP requests by putting subdirectory and host or address entries in the */etc/httpd.hostprefixes* file

- map virtual host addresses to the virtual host interface with the `ifconfig` command

Directing HTTP Requests

To direct HTTP requests, use the following format in the */etc/httpd.hostprefixes* file as shown in Table 8-1:

```
prefix [host-name-or-address ... ]
```

Table 8-1. HTTP Request Variables

Variable	Description
prefix	Specifies a subdirectory in the HTTP root directory, which is defined by the options httpd.rootdir command.
host-name-or-address	Specifies 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

About 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, or example, ***.gif***, ***.html***, or ***.mpg***, according to information in the */etc/httpd.mimetypes* file.

NOTE: On a Windows 9x 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 check box next to "Hide MS-DOS file extensions for file types that are registered."

Modifying MIME Content-Type Mappings

To modify MIME Content-Type mappings or to add MIME Content-Types, edit the entries in */etc/httpd.mimetypes*. Entries are in the following format:

```
# An optional comment.

suffix   Content-Type
```

Lines preceded by the # sign are comments. The file name suffix is case-insensitive. Following are sample entries:

```
# My clients' browsers can now use

# PICT graphics files.

pct         image/pict

pict        image/pict
```

In the sample entries, files whose names ended with ***.pact*** or ***.pact*** are mapped to the MIME Content-Type value of image/pict. 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.

Translating URLs

How the Filer Responds to 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 the configuration file contains up to three fields in the following format as shown in Table 8-2:

```
rule template result
```

Table 8-2. URL Response Fields

Field	Description
`rule`	Defines the response of the filer to a request.
`template`	Specifies a component of a URL.
`result`	Depends on the rule, as described in the following section.

Translation Rules Supported by the Filer

This section explains the meanings of the 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 within the HTTP root 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. The `result` field for the redirect rule must be specified as a complete URL beginning with *http://* and the host name.

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.

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 in */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 an */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.

For example, if the filer receives the following requests

```
http://filer/cgi-bin/displayfares

http://filer/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

Information in the /etc/log/httpd.log File

You can 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 /company.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

httpstat Statistic Types

The `httpstat` command displays four types of statistics about HTTP operations on the filer as shown in Table 8-3:

Table 8-3. httpstat Statistic Types

Column	Description
`gets`	Successful requests for files.
`badcalls`	Requests for nonexistent files.
`open conn.`	Number of HTTP connections currently open.
`peak conn.`	Largest number of simultaneous HTTP connections since the filer was booted or since the `-z` option was used.

Syntax

The syntax for the `httpstat` command is as follows:

```
httpstat [ -t|-z ][ interval ]
```

If you use no 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`:

```
httpstat
HTTPD statistics:
gets         badcalls         open conn.   peak conn.
451          11                5            17
```

CHAPTER 9
Snapshots

Understanding Snapshots

What Is a Snapshot?

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.

Accessing Snapshots

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 help, because files in snapshots can be viewed and copied by those who have permission to do so with the original files.

Simplifying Tape Backup

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.

Snapshots Use Little Disk Space

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."

Creating Snapshots for Your Needs

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.

Snapshots Maintain Original File Permissions

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.*

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.

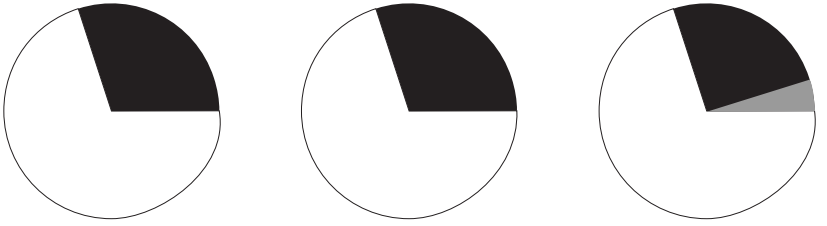
Example

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.

Diagram of a Snapshot

Figure 9-1 illustrates how disk space is used before and after *foo* is removed.



Before any snapshot is taken, disk space is consumed by the active file system only.

After a snapshot is taken, the active file system and snapshot point to the same disk blocks. The snapshot doesn´t use extra disk space.

After the *foo* file is deleted, the space previously used by *foo* is referenced by the snapshot. That´s why deleting active file system data doesn´t free disk space.

Space used by the active file system

Space used by the snapshot only

Unused disk space

Figure 9-1. Diagram of a Snapshot

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 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.

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.

Snapshot Commands and Options

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. Snapshot Commands

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 Dell custom MIB as described in "Using SNMP" in Chapter 4.

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. The options are shown in Table 9-2.

Table 9-2. Snapshot Options

Options	Descriptions
`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.*

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.

Types of Automatic Snapshots

Table 9-3 describes the three types of automatic snapshots.

Table 9-3. Automatic Snapshot Types

Type	Description
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.

Example 1 of snap sched Command

Figure 9- 2 shows a sample `snap sched` command:

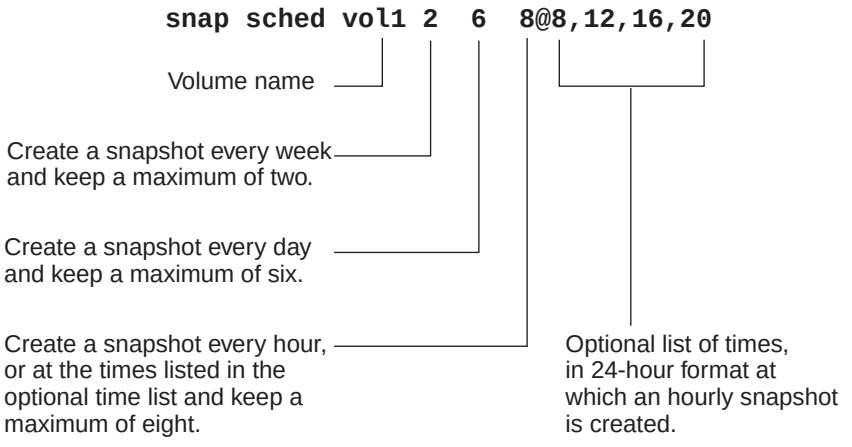


Figure 9-2. snap sched Command Sample

In the `snap sched` command, the first argument after the volume name in the example indicates how many weekly snapshots to keep (2), the second argument indicates how many nightly snapshots to keep (6), and the third argument indicates how many hourly snapshots to keep (8). 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 (8, 12, 17, 20). If the argument is omitted, the filer creates an hourly snapshot.

The default snapshot schedule is

`snap sched` *`volume_name`* `0 2 6@8,12,16,20`

Example 2 of snap sched Command

Following is an example of the `snap sched` command:

`snap sched` *`volume_name`* `2 6 8@8,12,16,20`

Snapshots Created by This Schedule

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
```

Result

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.

User-Defined Automatic Snapshots

You can create snapshots at predefined times instead of using the hourly, daily, and weekly schedules.

Example

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 filer snap create vol1 filename
```

NOTE: The `snap create` *command does not accept a snapshot name containing a slash (/).*

Understanding Snapshot Disk Consumption

About Snapshot Disk Consumption

It is important to understand the amount of disk space snapshots consume and the amount of disk space they are likely to consume. The following sections explain 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 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.

Using the df Command to Display Snapshot Use

To provide information about snapshot disk utilization, the `df` command on the filer treats snapshots as a partition different from the active file system.

Sample df command output

Following is a partial sample `df` command output:

```
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."

This section uses several examples to explain the advantages of reserving disk space for snapshots.

Snapshots Use Deleted Active File Disk Space

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 reassigned 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.

Administering Snapshot Disk Space

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%
```

Explanation

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.

Recovering Disk Space for File System Use

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.

Example

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" in Chapter 9 for information about how to use the `snap list` command to determine which snapshot to delete.

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.

Managing Snapshot Disk Consumption

About Snapshot Management

This section describes

- how to schedule snapshots to suit your environment
- how to determine a reasonable snapshot reserve
- 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 the `snap sched` command 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, Dell 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.
- When you create a new volume on a filer, 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.

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.

Command Output

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.

```
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

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 first number is equal to

$$\frac{\text{cumulative snapshot space} \times 100\%}{\text{cumulative snapshot space} + \text{file system space}}$$

- The second number is equal to

$$\frac{\text{this snapshot} \times 100\%}{\text{this snapshot} + \text{file system space}}$$

The %/Total Column

The %/total column shows space consumed as a percentage of total disk space in the volume.

- The first number is equal to

$$\frac{\text{cumulative snapshot space} \times 100\%}{\text{total disk space in this volume}}$$

- The second number is equal to

$$\frac{\text{this snapshot} \times 100\%}{\text{total disk space in this volume}}$$

"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).

Output Summary

The %/used number is more useful for planning the snapshot reserve because it is more likely to remain constant as the file system fills.

The example shows a volume that keeps five nightly snapshots and four hourly snapshots. 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, that is, 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 "Adjusting Disk Space Used by Snapshots" for further information about how to select a snapshot file for deletion 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 the following command:

`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.

Adjusting Disk Space Used by Snapshots

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," 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.

Example

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

About Client Access to Snapshots

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: NFS or CIFS.

NFS Client Access to Snapshots

Figure 9-3 shows the directory structure on a client with the *vol0* volume of a filer named filer mounted on */n/filer*.

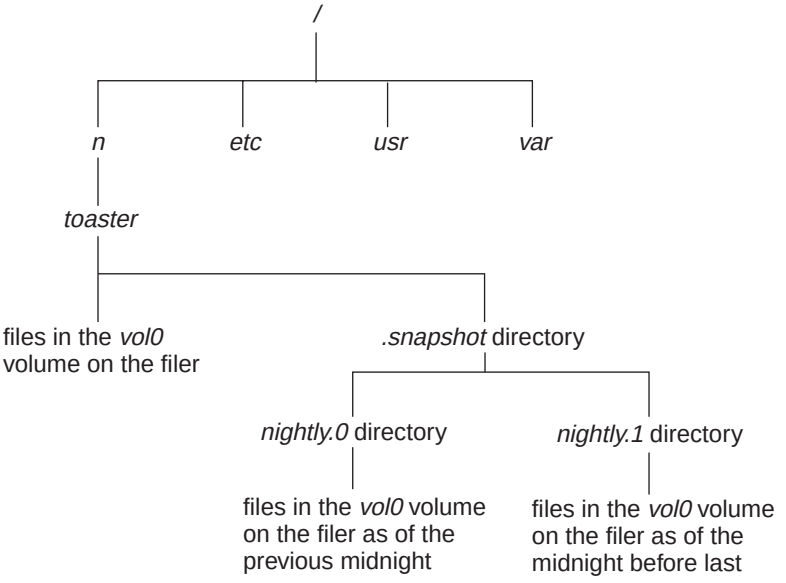


Figure 9-3. Directory Structure of NFS Client Access to Snapshots

Explanation

In this example, the client can obtain access to snapshots by way of */n/filer/.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"; that is, can be accessed when they are referenced by name but do 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.

CIFS Client Access to Snapshots

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 that 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 is displayed 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 drive F: 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.

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.

CHAPTER 10
qtree Administration

About qtrees

A qtree is a special subdirectory of the root directory of a volume.

qtree Parameters

You can set the following parameters on a qtree:

- security style: NTFS (Windows NT file system), UNIX, or mixed
- oplocks setting: On or Off
- disk space and file limits, as described in Chapter 11, "Quotas and Maximum Number of Files."

Volumes and qtrees

A volume has all the properties of a qtree except

- It can have qtrees under it.
- It has different defaults than a qtree.

Unless expressly mentioned otherwise, whatever applies to qtrees also applies to volumes.

NOTE: You cannot create a qtree inside another qtree.

Uses of qtrees

You use qtrees to group files that have similar characteristics, much in the way that you use volumes. However, qtrees are much more flexible than volumes.

Using qtrees

What You Can Do With qtrees

You can use qtrees in the following two ways:

- Group files that have the same security style and oplocks setting, such as files related to a particular activity, for example, a project, without having to create a separate volume for them.
- Perform quick and easy backups.

Using a qtree for a Project

One way to group files is to set up a qtree for a project, such as one maintaining a database. Setting up a qtree for a project enables you to do the following actions:

- Set the security style of the project without affecting the security style of other projects.

  For example, you use NTFS-style security if the members of the project use Windows files and applications. Another project in another qtree can use UNIX files and applications, while yet another project can use both Windows and UNIX files.

- Set oplocks (if the project uses Windows) as appropriate to the project without affecting other projects.

  For example, if one project uses a database that requires no oplocks, you can turn oplocks Off on that project's qtree. If another project uses oplocks, it can be in another qtree that has oplocks set to On.

- Limit the disk space and number of files available to a project so that it does not use up resources that other projects and users need. As the needs of the projects and available resources change, you can easily change the limits on the qtree. For instructions about managing space using qtrees, see Chapter 11, "Quotas and Maximum Number of Files."

Using a qtree for Backups

You can back up individual qtrees. You would do so to

- add flexibility to backup schedules
- modularize backups
- keep the size of each backup to one tape

For details, see Chapter 12, "data backup."

qtree and Volume Defaults

Volumes and qtrees have the default values shown in Table 10-1.

Table 10-1. qtree and Volume Defaults

Parameter	qtree default	Volume default
`oplocks`	On	On
`security`	The style of the volume's root directory	UNIX

Moving Files Between qtrees

In UNIX, you cannot move a file into or out of a qtree 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 qtree. You can always move the file by copying it, then deleting the original.

In Windows, you can move a file into or out of a qtree.

qtree Security Styles

Types of Security Styles

There are three kinds of security styles, described briefly in Table 10-2.

Table 10-2. qtree Security Styles

Style	Behavior
NTFS	Exactly like Windows NT NTFS: Files and directories have Windows NT file-level permission settings. *NOTE: To use NTFS security, make sure that the filer is licensed for CIFS.*
UNIX	Exactly like UNIX: Files and directories have UNIX permissions.
mixed	Both NTFS and UNIX security are allowed: a file or directory can have either Windows NT permissions or UNIX permissions.

qtree Security Styles in Detail

Table 10-3 describes the security styles in detail and the effects of changing to each style.

Table 10-3. qtree Security Styles in Detail

Security style	Description	Effect of changing to the style
NTFS	User access is determined as follows: **CIFS requests:** Windows NT permissions determine user access if Windows NT permissions have been set on a file. **NFS requests:** Windows NT permissions and a mapped CIFS identity determine access. UNIX groups are not used in the mapping from a UNIX identity to a CIFS identity\. *NOTE: You cannot change permissions in an NTFS qtree from a UNIX client.*	Windows NT permissions determine file access for a file that had them if the change is from a mixed qtree. Otherwise, UNIX-style permission bits determine file access for files created before the change.
UNIX	User access depends on the protocol, as follows: **CIFS requests:** Windows users are mapped to a UNIX UID and UNIX permissions determine access. In a UNIX qtree, a user cannot set Windows NT permissions. A Windows user can change UNIX permissions from Windows using SecureShare Access, as described in "Sending a Message to All Users on a Filer" in Chapter 7. **NFS requests:** Only the user's UID, GID, and UNIX-style permission bits of the file or directory determine user access.	The filer disregards any Windows NT permissions established previously.
mixed	Both NTFS and UNIX style permissions are permitted. The security style of a file is the style most recently used to set permissions on that file. See the NTFS information in "Types of Security Styles." **CAUTION: Changing NTFS permissions on a file recomputes UNIX permissions on that file.** **Changing UNIX permissions or ownership on a file deletes any NTFS permissions on that file.**	None.

qtree File Access Models

Kinds of File Access Models

You can use the following four file access models in working with qtrees:

- CIFS user accessing a file with Windows NT security
- CIFS user accessing a file with UNIX security
- NFS user accessing a file with Windows NT security
- NFS user accessing a file with UNIX security

CIFS Access to Windows Files

CIFS accesses to Windows files obey Windows security rules.

CIFS Access to UNIX Files

The following principles apply to accessing UNIX files from CIFS:

- All CIFS users are mapped to UNIX UIDs and GIDs.
- File accesses use UNIX security or PC security, as chosen during the `cifs setup` program.
  - UNIX-style permissions are determined by the rights associated with the UNIX UID and GID.
  - PC-style permissions are determined by the rights assigned in a share's Access Control List (ACL) and are limited by the UNIX permissions assigned to a file.
- PC security is like FAT (File Allocation Table) file system security with per-file permissions:
  - If the owner of a file or directory accesses an item, the owner permissions are checked to see whether they allow access.
  - If someone other than the owner of a file or directory accesses an item, the group permissions are checked to see whether they allow access.

NFS Access to Windows Files

The following principles apply to accessing Windows files from NFS:

- Windows NT permissions are mapped to UNIX permissions.
- Each Windows NT user who sets Windows NT permissions is mapped to a UNIX user and UNIX group, except that if the owner is a generic user, the owner is mapped to root with restrictions.
- Windows NT permissions for Owner are mapped to UNIX owner permissions.

- Windows NT permissions for Everyone are mapped to UNIX Group and UNIX Other permissions.

NFS Access to UNIX Files

NFS accesses to UNIX files obey UNIX security rules.

Creating a qtree

How to Create a qtree

To create a qtree, use the following command:

```
qtree create pathname
```

Result

The qtree *pathname* is created, with the following properties:

- Volume: the root volume, unless you specify another volume
- Name: *pathname*
- Security style: that of the root directory of the volume
- Oplocks settings: On

Creating a qtree in the Root Volume

If *pathname* does not begin with a slash (/), the qtree is created in the root volume. For example:

```
qtree create news
```

creates the qtree */vol/vol0/news*, where */vol/vol0/* is the default name for the root volume. For information about volumes, see "Volume Concepts" in Chapter 3.

Creating a qtree in a Volume Other Than the Root Volume

If you want to create a qtree (for example, *news*) in a particular volume (for example, *users*), use the following command:

```
qtree create /vol/users/news
```

Modifying the Security Style of a qtree

When to Change the Security Style of a qtree

There are many circumstance under which you might want to change qtree style. Two examples are

- Because the default security style of a qtree is that of its root volume, you might want to change the security style of a qtree after creating it to the style you want.
- You might also want to change the security style to accommodate other users or files; for example, if you start with an NTFS qtree and subsequently want to include UNIX files and users, you might want to change the qtree to a mixed qtree.

How to Change the Security Style of a qtree

To change the security style of a qtree, use the following command:

```
qtree security [pathname [mixed | ntfs | unix]]
```

Example With a qtree

To change the security model of */vol/users/docs* to be the same as Windows NT, use:

```
qtree security /vol/users/docs ntfs
```

Example With a Volume

To change the security model of the root directory of the *users* volume to mixed so that, outside of a qtree in the volume, one file can have NTFS security and another UNIX security, use

```
qtree security /vol/users/ mixed
```

NOTE: When you create an NTFS qtree or change a qtree to NTFS, by default, every Windows user is given full access. You must change the permissions if you want to restrict access to the qtree for some users. If you do not set NTFS file security on a file, UNIX permissions are enforced.

Modifying qtree Oplocks Settings

When to Change Oplocks Settings

You might want to change qtree oplocks settings when you add or remove software, or when the kind of data you are using changes. For detailed information about oplocks, see "Using Oplocks" in Chapter 7.

Changing Oplocks Settings

To change the oplocks setting of a qtree, follow these steps:

1. Make sure that the `cifs.oplocks.enable` option is set the way you want.
2. Use the qtree oplocks command, as follows:

```
qtree oplocks [name [enable | disable]]
```

The command takes effect immediately.

NOTE: If you disable the oplocks feature on a qtree, existing oplocks in the qtree are not broken.

Example With A qtree

To enable oplocks in the */vol/users/docs* qtree if oplocks are disabled and the `cifs.oplocks.enable` option is set to On, enter the following command:

```
qtree oplocks /vol/users/docs enable
```

Example With A Volume

To disable oplocks in the entire *users* volume if oplocks are enabled and the `cifs.oplocks.enable` option is set to On, enter the following command:

```
qtree oplocks /vol/users/ disable
```

This disables only files and directories that were not in a qtree when oplocks were enabled.

Effect of the cifs.oplocks.enable Option

Setting the `cifs.oplocks.enable` option has the following effects:

- If the `cifs.oplocks.enable` option is set to Off, all oplocks on the filer are turned off.
- If the `cifs.oplocks.enable` option is set back to On, the setting for each qtree comes into effect and oplocks are turned on for those qtrees where oplocks are enabled.

Displaying qtree Information

How to Display qtree Information

To display all attributes of all qtrees on a filer, use the `qtree` command with no arguments.

The qtree Command Display

The `qtree` command lists for a filer the items described in Table 10-4.

Table 10-4. qtree Command Display

Field	Contents
`Volume`	The volumes on a filer. Keep in mind that a volume is itself a qtree.
`qtree`	qtrees that are not volumes; each is listed next to its volume.
`Style`	The security style of each qtree.
`Oplocks`	The oplocks setting of each qtree.

Example qtree Display

For example:

`qtree`

```
Volume   qtree    Style Oplocks
-------- -------- ----- --------
bagels            unix  enabled
bagels   sesame   unix  enabled
muffin            ntfs  enabled
muffin   bran     unix  enabled
```

Explanation of Example qtree Display

In the example:

- Because *bagels* and *muffin* are volumes, each has a security style and oplocks setting.
- *Sesame* is a qtree in the *Bagels* volume. Its security style and oplocks setting happen to be the same as that of its parent volume, bagels.
- *Bran* is a qtree in the *muffin* volume. Its security style is different from that of its parent volume, *muffin*. Files in *bran* have UNIX-style permissions; files in *muffin* but not in *bran* have NTFS-style permissions.

CHAPTER 11
Quotas and Maximum Number of Files

Restricting or Tracking Disk Usage by Using Disk Quotas

About Disk Quotas

Filer disk quotas restrict disk space and the number of files used by a user, a group, or a qtree. For information about how to create a qtree, refer to Chapter 10, "qtree Administration." This chapter discusses how to manage disk quotas by editing the */etc/quotas* file.

Format of the Quotas File

To set up disk quotas, create a ***quotas*** file in the */etc* directory.

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.*

Following is a sample ***quotas*** file:

```
#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: If the quota is a tree quota, the field in the `type` *column of the quotas file displays* `tree`*, not* `qtree`*.*

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

Specifies the user, group, or qtree 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 qtree. 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 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 qtree. 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 to the entire volume, specify `group@tree` where *tree* is the name of a qtree. 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

Specifies the maximum amount of disk space that the quota target can use. In this field, K is equivalent to 1,024 bytes, M means 2^20 bytes, and G means 2^30 bytes. If you omit the K, M, or G, the default is K.

If you want to track the disk usage but do not want to impose a disk usage limit on the quota target, enter a dash (-) in the disk field.

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

Specifies the maximum number of files that the quota target can use. Use K to indicate 1,024 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.

If you want to track the number of files but do not want to impose a limit on the number of files that can be used by the quota target, enter a dash (-) in the files field.

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
mfisher                user                   -        -
```

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 qtree 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 qtree provided that the quotas on the proj1 qtree 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 qtree, 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 qtree, 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 qtree, mhoward can use 75 MB of disk space and 76,800 files.

- There is no limit as to how much disk space or how many files the user mfisher can use. You can, however, use the `quota report` command to display the amount of disk space and the number of files used by this user.

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

Some of the information from `quota` commands is available through SNMP using the Dell custom MIB. For more information about the MIB, refer to "About the Dell Custom MIB" in Chapter 4.

Enabling or Disabling Quotas

The `quota on|off` command enables or disables quotas for all volumes or a specific volume. Use the following syntax when using the command:

```
quotas [on|off] volume
```

Use the following command 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, use the `quota` command without any arguments. For example:

```
quota

vol0: quotas are on.
vol1: quotas are initializing (24% done).
vol3: quotas are off.
```

Use the following command 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."

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.

Creating an Active Quota for a Quota Target

You can 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 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
   ```

   Substitute the values you want for 300M and 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
   ```

   Substitute the values you want for 700M and 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
   ```

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, enter the following commands:

   ```
   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 qtree. If the write would exceed the qtree quota, the filer logs the following error message:

```
tid tree_id: tree quota exceeded on volume_name
```

If the qtree 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/filer* on a client named client2:

```
rlogin client2
You have mail.
Block limit reached on /t/filer
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/filer/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

About 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.

Viewing the Number of Files in a Volume

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:

```
df -i /vol/home

Filesystem     iused     ifree    %iused   Mounted on
/vol/home/      2872      118090      2%    /vol/home

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 Dell custom MIB. For more information about the MIB, see "About the Dell Custom MIB" in Chapter 4.

The df Command

About 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 Dell custom MIB, which is described in "About the Dell Custom MIB" in Chapter 4.

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:

```
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.

NOTE: 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.*

Using the df Command With qtrees

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/filer*, the `df` command on the client displays the disk information about the */t/filer* file system as follows:

```
df /t/filer/engineering/jdoe

Filesystem            kbytes    used   avail capacity  Mounted on
filer:/            2097151 1646923  450228    79%  /t/filer
```

However, if you defined qtrees on the filer, the information about available space could be misleading because the actual space available might be less. For example, if */engineering* is a qtree with a disk quota of 1,800 MB, the space available in the */t/filer/engineering* directory is less than that in the `df` command output shown in the preceding example.

If you have qtrees on the filer, Dell recommends that you mount each qtree separately. For example, if the filer named filer has two qtrees, */vol/home/engineering* and */vol/home/marketing*, mount *filer:/vol/home/engineering* and *filer:/vol/home/marketing* on two mount points, for example, */t/filer/engineering* and */t/filer/marketing*.

In this way, the filer takes the qtrees into consideration when responding to a `df` command from a client and returns the amount of free space in each qtree, as opposed to the space available in the entire file system.

CHAPTER 12
Data Backup

Introduction to Data Backup

Meaning of Data Backup

Data backup means copying data from disk to tape. While the filer dump command enables you to copy data to standard output, this chapter mainly discusses how to copy data to tape.

Why You Want to Back Up Data From Disk to Tape

The following list describes the reasons for backing up data from disk to tape:

- You can restore data from tape if an application or a user inadvertently corrupts or deletes files that cannot be recovered from snapshots.
- You can store the backup tapes at an off-site archive to protect the data against natural disasters.
- After you reinstall the file system on the filer (for example, for migrating to larger disks or for converting a single-volume filer to a multivolume filer), you can restore data from tape.

Different Methods for Backing Up the Filer

The filer supports the `dump` command backup method.

CAUTION: If you use the `tar` or `cpio` command on an NFS client to back up the filer, be aware that some versions of these commands fail with file systems that contain long path names, unusual file names, or hard links.

How the dump Command Works

Purpose of the dump Command

The dump command writes file system data from disk to tape in a format that enables you to restore the data to a filer using the filer's restore command or the Solaris ufsrestore command.

What the dump Command Can Back Up

The dump command can back up a file, a directory, a qtree, or an entire volume. In the dump command, you specify the complete path name to be backed up. In this chapter, this path name is referred to as the "dump path."

How the dump Command Uses Snapshots to Back Up Data

The dump path can exist in the filer's active file system or in a snapshot. If the dump path is in an active file system, the filer takes a snapshot of the active file system before it writes the data to tape. The snapshot capability ensures that the data written to tape is consistent. As a result, you need not take the filer or volume off-line before initiating the backup.

The dump command names each snapshot it creates *snapshot_for_backup.n*. The *n* at the end of the snapshot name is an integer starting at 0. Each time the dump command creates a snapshot, it increments the integer by 1. The filer resets the integer to 0 when it is rebooted.

The dump command automatically deletes the snapshot after it successfully finishes the backup.

When the filer executes multiple dump commands simultaneously, the dump commands create multiple snapshots. For example, if the filer is running two dump commands simultaneously, you find these snapshots in the volumes from which data is being backed up: *snapshot_for_backup.0* and *snapshot_for_backup.1*.

CAUTION: When the dump command is in progress, the filer does not allow you to delete the *snapshot_for_backup.n* file. If you are backing up data from an hourly, daily, or weekly snapshot, make sure that the snapshot scheduler does not delete the snapshot before the dump command is finished.

Metadata Being Backed Up

In addition to backing up data within files, the dump command backs up these types of metadata:

- UNIX group ID, owner ID, and file permissions
- UNIX access time and modify time

- File type, including UNIX symbolic links and hard links
- File size
- DOS name, attributes, and create time
- Windows NT ACLs

NOTE: The CIFS attributes (DOS name, attributes, and create time, and Windows NT ACLs) can be restored only with the filer's `restore` *command. You cannot use the Solaris* `ufsrestore` *command to restore these attributes, although the* `ufsrestore` *command can restore the data in CIFS-created files.*

How to Exclude Certain Types of Data From the Backup

Options in the `dump` command enable you to exclude certain types of data from the backup.

Windows NT ACLs

You can choose not to back up Windows NT ACLs if the data is used only by NFS clients.

Exclude List

You can choose to exclude files and directories from a backup if you do not need those files and directories again. For example, you can exclude temporary files generated by some applications or object files produced during program compilation.

When the `dump` command traverses the dump path, it compares file and directory names to each exclude string specified in the `dump` command. If the name exactly matches the string, the file is excluded from the backup.

NOTE: You can reduce the amount of backup data by using an exclude list in the `dump` *command. However, an exclude list increases the amount of time needed to finish a* `dump` *command because the filer must compare each file name to the exclude list to determine whether the file should be backed up.*

Devices Used by the Dump Command

The `dump` command can back up data to these devices:

- Tape drives or tape stackers attached to the filer
- Tape drives or tape stackers attached to another computer, provided that the following requirements are met:
  - The computer supports the `rmt` protocol.
  - The filer has a trusting relationship with the remote computer to which the tape drive is attached. This relationship enables the filer to write to the tape drive. For example, if you want to back up the filer to a tape drive attached to another filer, include the filer in the */etc/hosts.equiv* file of the destination

filer. If you want to back up the filer to a tape drive attached to a SunOS or Solaris computer, include the filer in the */.rhosts* file on the computer.

- The filer can resolve the name of the computer to which the tape drive is attached using the information about the computer in the filer's */etc/hosts* file or in the DNS database.

- Standard output, provided that you enter the `dump` command through `rsh`. Because the console is not a standard output device, you cannot write to standard output if you enter the `dump` command on the console.

Incremental Backups

You can specify the level of backup in a `dump` command ("dump level"), which ranges from level 0 to level 9. A level 0 backup is a full backup: It writes all data in the dump path to the backup media. Backups at dump level ranging from level 1 to level 9 are incremental backups. In an incremental backup, only files changed since the previous level are written to the backup media.

Where to Enter the Dump Command

You can enter the `dump` command through the console or through `rsh`.

Benefits of Entering the dump Command Through rsh

Entering the `dump` command through `rsh` gives you these benefits:

- When the `dump` command is in progress, you can still use the console to manage the filer. If the `dump` command entered on the console is backing up a large number of files, you cannot use the console for a long time.
- You can start multiple `dump` commands through `rsh`.
- It is less likely to inadvertently terminate the `dump` command. If you enter a `dump` command on the console, it could be terminated by a Ctrl-C entered on a host connected to the filer using `telnet`.
- You can automate filer backups through shell scripts.
- You can write data to standard output.

Benefits of Entering the dump Command on the Console

If you enter the `dump` command on the console, you can read and respond to screen messages displayed by the command. For example, the command might prompt you for another tape to complete the backup. A `dump` command entered through `rsh` terminates and does not generate any messages when the command needs user intervention.

Format of the Backup Data

About This Section

When the filer executes the `dump` command, it displays messages showing the different passes of the `dump` command. This section discusses the format of the backup data and what data is written to tape in each pass of the `dump` command.

Backup Data Format

The backup data format is organized based on inodes. An inode for a file or a directory contains information for tracking the file's or the directory's type, time stamps, bad blocks, and so on.

On each tape, the `dump` command creates two maps:

- The first map shows which inodes are used in the directory to be dumped. The filer uses this map to determine which files have been deleted or moved between incremental dumps.
- The second map shows which inodes have been written to the tape. The filer uses this map to verify the accuracy of the restore operation when the backup data is restored.

The `dump` command writes files and directories to tape by inode numbers.

Five Passes of the dump Command

The `dump` command consists of five passes. After you enter the `dump` command, the filer displays messages showing which pass is in progress.

In passes 1 and 2, the filer traverses directories to search for files to be backed up. The filer backs up a file if the file meets these requirements:

- The file is included in the dump path.
- The file has changed since the previous backup at a lower dump level.
- The file is not excluded by the exclude list specified in the `dump` command.

Example

If you initiate a level-1 `dump` command to back up */vol/vol0*, the filer searches for files in */vol/vol0* that have changed since the previous level-0 backup.

In passes 1 and 2, the filer also creates the maps described in "Format of the Backup Data."

In passes 3 and 4, the filer writes the data to tape in ascending inode order.

In pass 5, the filer writes the ACL information to tape.

How the dump Command Writes and Stores Data on Tape

About This Section

The dump command transfers a number of blocks of data at a time to an output file on tape. This section provides information about how the command writes and stores these blocks of data, which helps you decide the dump command format appropriate for your backup.

Meaning of Tape Block

A tape block is 1 kilobyte of data. In the dump command, you can specify the number of tape blocks that are transferred in each write operation. This number is called the "blocking factor."

CAUTION: If you plan to restore the backed-up files on a computer other than the filer, make sure that the blocking factor you choose does not exceed the maximum blocking factor supported by that computer. If you use a filer to restore the backup data, the blocking factor must not exceed 63.

Meaning of Tape File

A tape file is a dump output file on tape. It can also be an entire tape. You can back up a dump path to one tape file on a tape, multiple tape files on a tape, or multiple tape files on multiple tapes.

In the dump command, you can specify the maximum size of the tape file in terms of tape blocks. For example, if you want the maximum tape file to be 2 GB, specify 2,097,151. (That is, the largest tape file can contain 2,097,151 tape blocks, which are 1 kilobyte each.)

CAUTION: If the backed-up files are to be restored on a computer other than the filer, make sure that the tape file size you choose does not exceed the maximum tape file size supported by that computer. If you use a filer to restore the backup data, the tape file size can be as large as a single tape.

When the Dump Command Writes to Multiple Tape Files

When the data being backed up exceeds the capacity of a tape, the dump command automatically writes to the next tape file specified in the command. If the current tape file is the last tape file listed on the dump command, the filer prompts you to load another tape.

If you specify the maximum tape file size in the dump command, the command writes to the next tape file when the backup data reaches that size, regardless of the amount of space left in the current tape.

A tape can contain multiple tape files, but any tape must contain at least one complete tape file.

Different Types of Tape Files

The `dump` command can use these types of tape files:

- Local or remote. (Refer to "Devices Used by the Dump Command" for more information about where the tape devices can be located.)
- Rewind or norewind. If the tape file is a rewind file, the filer rewinds the tape after it finishes writing the tape file. If you want the `dump` command to write multiple tape files on the same tape, specify norewind tape files in the `dump` command.

  The name of a norewind tape file begins with an `n` (for example, `nrst0a`); the name of a rewind tape file begins with an `r` (for example, `rst0a`).

- Unload/reload tape files. These tape files apply only if you are backing up to tapes in tape stackers. When you use these tape files, the `dump` command unloads the tape when it reaches the end of tape, and then the tape stacker reloads another tape.

  The name of an unload/reload tape file begins with a `u` (for example, `urst0a`).

  An unload/reload tape file cannot be a norewind tape file. For example, do not specify `unrst0a`.

NOTE: If the tape stacker is attached to a remote host, the `dump` *command can reload the tape only if that host supports automatic reloading.*

- Standard output, which is specified as - in the `dump` command.

Determining the Amount of Backup Data

Description

Before you enter the `dump` command, you must determine the amount of backup data so that you can estimate the number of tape files and the number of tapes required for the backup.

The procedure for estimating the amount of data depends on whether the data is in a qtree.

Step for Estimating the Amount of Data If You Back Up A qtree

To display the number of kilobytes used for the qtree, enter the `quota report` command.

Steps for Estimating the Amount of Data if You Back Up Data Not In A qtree

The procedure for estimating the amount of backup data depends on whether the filer is mounted on an NFS client or is shared by a CIFS client.

If the Filer Is Mounted on an NFS Client

Follow these steps to determine the amount of backup data from an NFS client:

1. On the NFS client, change directory to the mount point.
2. Enter the following command for each directory you want to back up:

   `du -s pathname ...`

   Example: If the NFS client mounts the filer to */filer* and you want to back up the */etc* and */home* directories, enter these commands:

   ```
   cd /filer
   ```

   ```
   du -s etc home
   ```

   The command output shows the amount of space allocated for the directories. Refer to the documentation for your client system for interpreting the output because the output is written in different units (for example, 512-byte units or 1,024 units) depending on the operating system.

If the Filer Is Shared by a CIFS Client

Follow these steps to determine the amount of backup data from a CIFS client:

1. On the CIFS client, point to the shared file or directory that you want to back up.
2. Right-click to display the pull-down menu.
3. Click Properties to display the number of bytes used by the file or directory.

Determining the Number of Tapes for the Backup

Description

You must determine the number of tapes required for the backup before entering the `dump` command for these reasons:

- You can ensure that the `dump` command will not run out of tapes and be incomplete.
- You can load all tapes required in the tape drives or stackers in advance for an unattended backup. If you do not load enough tapes before entering the `dump` command and you start the `dump` command from the console, the filer prompts

you to load additional tapes. If you start the `dump` command through `rsh`, you do not see the prompts from the filer and the `dump` command cannot be completed because of the lack of tapes.

Prerequisites

You must meet these prerequisites before you can determine the number of tape files required:

- The filer must be able to display tape drive information. To learn how to display tape drive information, refer to "Displaying Tape Device Information" in Chapter 14.
- You must know the amount of data to be backed up. For information about determining the amount of backup data, refer to "Determining the Amount of Backup Data."

Steps

Follow these steps to determine the number of tapes required for the backup:

1. Determine the capacity of the tape drives you are using for the backup.
2. Determine the amount of data that needs to be backed up.
3. Determine the amount of space that will be left unused on a tape. For example, if the `dump` command specifies several tape files, the command automatically writes to the next tape file even though there is space left in the current tape.

   Example: The following `dump` command starts the second tape file on `rst1a` after writing 2,000,000 tape blocks to the first tape file, even though the tape in `rst0a` contains unused space:

   ```
   dump 0ufB rst0a,rst1a 2000000 /vol/vol0
   ```

Prerequisites for the dump Command

About This Section

The prerequisites for the `dump` command depend on the format of the `dump` command. For example, the prerequisites for backing up data to a remote tape drive and to a local tape drive are different. This section describes the general prerequisites that must be met regardless of the `dump` command format. It also describes the specific prerequisites for specific backup procedures.

General Prerequisites

You must meet these prerequisites for the `dump` command to run successfully:

- You must have enough tape space for the backup. Refer to "Determining the Number of Tapes for the Backup" to determine the number of tapes required for the backup.
- The tape drive is available. That is, it is not currently used for data backup or recovery.

Prerequisites for Backing Up to a Nonqualified Tape Drive

The tape drive must be included in the */etc/cloned_tapes* file.

Prerequisites for Backing Up to a Remote Tape Drive

These are the prerequisites for backing up data to a tape drive attached to a remote host:

- You must know the maximum blocking factor supported by the remote host to which data is backed up. Some computers allow a blocking factor greater than the default used by the `dump` command (63). Make sure that the blocking factor you specify in the `dump` command does not exceed the maximum blocking factor supported by the remote host.
- If you use a blocking factor greater than 63, you must know the maximum blocking factor supported by the system that you plan to use for restoring the data. If the maximum blocking factor is 63 on the system for restoring data, that system cannot restore data from a tape file created with a blocking factor greater than 63.
- You must know the maximum tape file size supported by the remote host to which data is backed up. For example, some UNIX systems do not support tape files larger than 2 GB. In the `dump` command, you need to specify the tape file size appropriate to your remote host.
- If you use a tape file size larger than 2 GB, you must know the maximum tape file size supported by the system that you plan to use for restoring the data. If the maximum tape file size is 2 GB on the system for restoring data, that system might not be able to restore data from a tape file greater than 2 GB.
- The remote host must support the `rmt` protocol.
- The filer being backed up must have a trusting relationship with the remote host.

Recommendations for Performing a Backup

About This Section

Some recommendations in this section are applicable to all kinds of backups; some recommendations depend on your priorities, such as your need to minimize backup time or to minimize tape handling.

General Recommendations

Follow the recommendations in this section when you back up data from the filer.

Avoid Backing Up Too Much Data in a Single Dump Command

The reasons are listed below:

- The `dump` command cannot be restarted. That is, if the `dump` command encounters an error, you cannot correct the error and proceed from the point where the command fails. You must start the command from the beginning.
- The backup takes a long time to finish. This leaves you with a long time period during which changed data cannot be written to tape by an incremental backup.

Store Incremental Backups for the Same Dump Path on the Same Tape

If you want to do a subtree restore, you must restore from all incremental backups to restore the most recent data. Having all the incremental backups in the same tape minimizes tape management during the restore process.

Write Down Qtree Information Before Backing Up qtrees

Record the information about qtrees if your backup contains qtrees. This is because when you restore data from tape, the filer does not automatically re-create the qtrees. You must re-create the qtrees into which data is to be restored. Having a record of all qtrees enables you to re-create the qtrees quickly.

Recommendations for Minimizing Backup Time and Data Loss

The recommendations in this section enable you to minimize both the time required to perform a backup and the possibility of data loss. This is because the shorter the time for the `dump` command to finish, the more incremental backups you can perform. A large number of incremental backups minimize the amount of unrecoverable data.

NOTE: There is a disadvantage to having a large number of incremental backups: When you restore data, you must restore from all the incremental backup tapes, which might take a long time.

Use Multiple Local Tape Drives

Attach the maximum number of tape drives to the filer. The filer can write faster to a local tape drive than to a tape drive attached to another system.

Organize Data to be Backed Up

The first two passes of the `dump` command run faster if the dump path is one of these:

- A volume
- A qtree
- Data not belonging to any qtrees

Limit the Amount of Data in Each Backup

Limit the amount of data in a volume or qtree to be backed up to 200 GB.

Schedule the Backups Appropriately

Schedule backups when the load on the filer is light. Refer to "Example of Backing up the Entire Filer" for an example of executing one full backup and several incremental backups a night.

Avoid Using an Exclude List

If you exclude certain files from the backup, the `dump` command takes longer because it must compare each file name to the strings in the exclude list to determine whether it should back up the file.

Recommendations for Minimizing Downtime During Data Recovery

The method used for restoring data has a greater effect on downtime than does the `dump` command. However, it helps to keep the data in the dump path to 100 GB or less because a full restore is more efficient than a partial restore. For example, it takes less time to restore an entire 100-GB qtree than to extract 100 GB of data from a 500-GB backup.

Recommendations for Minimizing the Number of Tape Drives Required

Because the filer supports the `rmt` protocol, several filers can share the same tape drive. Attach the tape drive to the filer with the most data to back up. Follow these guidelines if multiple filers back up to the same tape drive:

- Use a private network for the backup so that the traffic load on the network does not slow down the backup process.
- Schedule the `dump` command on each filer so that it starts only when no other filers are using the tape drive.

The dump Command Syntax

Command Syntax

The `dump` command syntax is as follows:

```
dump [ option argument ] path
```

Rules for Entering the dump Command

The following list describes the rules for entering the `dump` command:

- *option* can be a list. You must list all options together; do not separate the options by commas or spaces.
- *argument* can be a list of arguments, each of which is associated with an option. The arguments are separated by spaces.
- If an option takes a list of arguments, the arguments in the list are separated by commas.
- List the arguments in the same order as you list the options.
- The dump level must be the first option. You can type other options in any order.
- *path* is the complete path name of the file or directory to be backed up by the `dump` command.
- Always precede the volume name by `/vol/` even though the volume is a root volume. This is because between different levels of backups, you might have changed the root volume.

NOTE: Some options do not have any arguments.

Example of a Simple dump Command

The following example illustrates how you use the `dump` command syntax:

```
dump 0ufb rst0a 63 /vol/vol0/
```

Options

The following list describes the options:

- `0` is the dump level.
- `u` updates the */etc/dumpdates* file.
- `f` specifies the tape file.
- `b` specifies the blocking factor.

Arguments

The `0` and `u` options do not have arguments.

The argument for the `f` option is `rst0a` (the tape file name), and the argument for the `b` option is `63` (blocking factor). The `rst0a` and `63` arguments must be listed in the same order as the `f` and `b` options.

Path

The path name is */vol/vol0/*, which means that all files and directories in the *vol0* volume are backed up to tape.

NOTE: The `dump` *command consists of more options than those described in this section. All options are described in greater detail in the next section.*

Descriptions of dump Options

Table 12-1 describes the meanings of the `dump` command options.

Table 12-1. dump Command Descriptions

Option	Meaning
Dump level	It is mandatory. It can be a number from 0 to 9. Level 0 is a full backup; levels 1 through 9 are for incremental backups.
`A`	The `dump` command does not back up Windows NT ACLs.
`b`	It is the blocking factor. It takes a number as an argument, which is the number of 1-KB blocks in each write. If you use a local tape drive for backup, the number should be from 4 to 64. If you use a remote tape device for backup, the number must be from 4 to 256. The `dump` command generates an error message if you specify a number that is out of range for your tape drive. The default argument is 63.
`B`	It takes a number as an argument, which is the number of tape blocks in a tape file. The `dump` command writes the specified number of tape blocks to a tape file before starting a new tape file. The argument must be equal to or larger than the argument to the `b` option. If you do not specify this option, the `dump` command writes data until it reaches the end of tape.
`f`	It is mandatory. It takes the tape file name as an argument. You can specify a comma-separated list of tape file names.
`l`	It backs up only specific files and directories in the dump path. It must be used with the `n` option.
`n`	It takes a string as an argument, which is the name of the backup to be recorded in the */etc/dumpdates* file. It is required if you use the `l` option.

Table 12-1. dump Command Descriptions *(continued)*

Option	Meaning
Q	It takes a volume name as argument. The `dump` command backs up all data in the specified volume that does not reside in a qtree. You cannot perform incremental backups on data that is backed up with the `Q` option. However, you can use both the `Q` option and `u` option to record the backup in the */etc/dumpdates* file. The entry in the */etc/dumpdates* file enables you to keep a history of the backups.
u	The `dump` command updates the */etc/dumpdates* file, which contains the dump path, the dump level, and the creation time of the snapshot used by the `dump` command. You must use this option if you plan to perform incremental backups in the future.
X	It takes a string as argument. You can specify a comma-separated list of strings. If the name of a file in the dump path matches one of the strings, the `dump` command excludes that file from the backup. Each string for the `X` option applies to files at every directory under the dump path. The following list describes the rules for specifying the strings: • To exclude a file, the name of the file must match the string exactly. For example, if you specify `core`, only those files whose names are `core` are excluded. A file named `a.core` is not excluded. • You can use the asterisk (*) as a wildcard character. • The wildcard character must be the first or last character of the string. Each string can contain up to two wildcard characters. For example, you can specify `*.core`, `core.*`, or `*core.*`, but not `core*.1`. • Because the strings in the list are comma-separated, if you want to exclude files whose names contain a comma, precede the comma in the string with a backslash. • You can specify up to 32 strings for the `X` option.

Using the dump Command to Back Up Data to Tape

Description

You can enter the `dump` command at any time to back up data in a specified path. After the `dump` command is finished, the data in the path is written to tape.

Prerequisites

You must meet all the prerequisites described in "Prerequisites for the Dump Command."

Restrictions

You can run up to four `dump` commands in parallel.

Steps

Follow these steps to execute a `dump` command:

1. Prepare the number of tapes required, following the steps described in "Determining the Number of Tapes for the Backup."
2. Determine and note the blocking factor used in the backup. For more information about blocking factors, refer to "Meaning of Tape Block."
3. Determine and note the tape file size used in the backup. For more information about tape file size, refer to "Meaning of Tape File."
4. Enter the `dump` command in the appropriate format, using the examples that follow as a guide.

Examples of Level-0 Backups to a Local Tape File

The following list provides examples of level-0 backups to a local tape file.

- **`dump 0f rst0a /vol/vol1/users/tom/specs`**

  The command performs a full backup of the */vol/vol1/users/tom/specs* directory. After the `dump` command is finished, the filer rewinds the tape.

- **`dump 0uf rst0a /vol/vol1/users/tom`**

  The command performs a full backup of the */vol/vol1/users/tom* directory and records the backup in the */etc/dumpdates* file.

- **`dump 0fQ rst0a /vol/vol1`**

  The command performs a full backup of all data in the */vol/vol1* volume that does not belong in any qtrees.

Examples of Backups to a Remote Tape File

- **`dump 0f filer1:nrst0a /vol/vol1`**

  The command performs a backup to a tape drive attached to a filer named filer1.

- **`dump 0f unix_machine:/dev/rst0 /vol/vol1`**

  The command performs a backup to a tape drive on a UNIX system.

Example of an Incremental Backup to a Local Tape Drive

- **`dump 1uf nrst0a /vol/vol1`**

  The command performs a level-1 backup of the */vol/vol1* volume to `nrst0a`. This means that only files that have changed since the most recent level-0 backup are written to tape. After the command is finished, it does not rewind the tape.

Examples of Backups to Multiple Tape Files

- **`dump 0f rst0a,rst1a /vol/vol1`**

  The command backs up the */vol/vol1* volume to the tape on `rst0a`. If the volume exceeds the capacity of the tape on `rst0a`, the command writes the second tape file to the tape on `rst1a`. Otherwise, it does not use the tape on `rst1a`.

- **`dump 0f urst0a,urst0a /vol/vol1`**

  The command backs up the */vol/vol1* volume to the tape in `urst0a`. After the command finishes writing to the first tape drive, it does not rewind the tape. If the volume exceeds the capacity of the tape on `urst0a`, the command writes the second tape file to the same tape drive on `urst0a`. If the command needs to write the third tape file, it prompts you to load a new tape.

NOTE: In this example, if you know in advance that the backup requires two tape files, use `rst0a` *as the second tape file name in the* `dump` *command so that the command automatically rewinds the tape after it is finished.*

Example of Backing Up a Directory From a Snapshot

- **`dump 0f rst0a /vol/vol1/.snapshot/weekly.0/home/users`**

  From the weekly snapshot, the command performs a level-0 backup of the */vol/vol1/home/users* qtree.

Example of Backups to a Tape Stacker

- **`dump 0f urst0a,urst0a,urst0a,nrst0a /vol/vol1`**

  The command backs up the */vol/vol1* volume to a tape stacker. After the command finishes writing a tape file, it unloads the tape and reloads another tape. After the command finishes writing the fourth tape file, it does not rewind the tape.

Example of Backing Up Multiple Files or Directories in One dump Command

- **`dump 0ufnl rst0a user.1.3.5 /vol/vol1/home`**

  The `n` option specifies the name of the backup (`user.1.3.5`), which is recorded in the */etc/dumpdates* file. If you do not specify a name, the `dump`

command fails. The `l` option specifies that you interactively enter the names of individual files and directories to be backed up from the */vol/vol1/home* directory.

The filer displays some messages and a prompt for the names of the files and directories you want to back up. Enter each name as a path name relative to the dump path in the `dump` command. Do not specify ".." or specify a directory that contains symbolic links. To end the list of names, use a blank line or press Ctrl-D. Then the `dump` command displays messages about the progress of each pass of the command.

The following example shows how to enter the relative path names for the directories to be backed up. The example ends the list of path names with a blank line.

```
DUMP: creating "snapshot_for_backup.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
user1
user3
user5
```

Example of Backing Up Data Without ACLs

- **`dump 0uAf rst0a /vol/vol1`**

  The command performs a level-0 backup of the */vol/vol1* volume. The `A` option means that the backup does not include any Windows NT ACL information. Use this option if the files in the volume are for NFS only.

Example of Specifying a Blocking Factor

- **`dump 0ufb rst0a 32 /vol/vol1`**

  The command performs a level-0 backup of the */vol/vol1* volume. The command writes 32 KB of data at a time, enabling you to restore the data from systems that limit each write to 32 KB.

Example of Specifying a Tape File Size

- **`dump 0ufB nrst0a,nrst0a 2000000 /vol/vol1`**

  The command performs a level-0 backup of the */vol/vol1* volume to the tape in `nrst0a` using a tape file size of 2,000,000 tape blocks. After writing 2,000,000 tape blocks to the first tape file, the command does not rewind the tape; it continues writing the second tape file on the same tape. If there is more data to be backed up after the command reaches the end of the second tape file, the command prompts you for a new tape.

Example of Excluding Files From a Backup

- **`dump 0ufX rst0a tmp,*.o,core*,*backup*,usr\, /vol/vol1`**

  The command performs a level-0 backup of the */vol/vol1* volume, which excludes the files that meet one of these requirements:

  — The name is `tmp`.

  — The name ends in `.o` (for example, `program.o`).

  — The name begins with `core` (for example, `core.small`).

  — The name contains `backup` (for example, `spec.backup.1`).

  — The name is `usr,`.

Example of Backing Up to a Tape Stacker Shared by Multiple Filers

This section provides a sample schedule for backing up the following filers to the same tape stacker:

- filer1, which contains 3 volumes
- filer2, which contains 2 volumes
- filer3, which contains 2 volumes

The schedule uses the following assumptions:

- The level-0 backup takes 10 hours.
- The tape stacker is attached to filer1, and the tape stacker contains two tape drives.
- Root on each filer has the permission to write to filer1

According to this schedule, the filers perform the following types of backups:

- The filers fully back up each volume once a week.
- The filers perform a level-1 backup and a level-2 backup for each volume every week.
- The filers do not rewind the tape for each level-1 backup so that the subsequent level-2 backup can be written immediately after the level-1 backup on the tape.

Table 12-2 shows the sample backup schedule:

Table 12-2. Sample of Backing Up to Tape Stacker Shared by Multiple Filers

Day of week	dump commands on each filer
Sunday	filer1: **dump 0uf urst0a,urst0a,urst0a /vol/vol1** filer3: **dump 1uf filer1:nrst1a /vol/vol6** filer2: **dump 2uf filer1:urst1a /vol/vol4**
Monday	filer1: **dump 0uf urst0a,urst0a,urst0a /vol/vol2** filer3: **dump 1uf filer1:nrst1a /vol/vol7** filer2: **dump 2uf filer1:urst1a /vol/vol5**
Tuesday	filer1: **dump 0uf urst0a,urst0a,urst0a /vol/vol3** filer1: **dump 1uf nrst1a /vol/vol1** filer3: **dump 2uf filer1:urst1a /vol/vol6**
Wednesday	filer2: **dump 0uf filer1:urst0a,urst0a,urst0a /vol/vol4** filer1: **dump 1uf nrst1a /vol/vol2** filer3: **dump 2uf filer1:urst1a /vol/vol7**

Table 12-2. Sample of Backing Up to Tape Stacker Shared by Multiple Filers *(continued)*

Day of week	dump commands on each filer
Thursday	filer2: **`dump 0uf filer1:urst0a,urst0a,urst0a /vol/vol5`** filer1: **`dump 1uf nrst1a /vol/vol3`** filer1: **`dump 2uf urst1a /vol/vol1`**
Friday	filer3: **`dump 0uf filer1:urst0a,urst0a,urst0a /vol/vol6`** filer2: **`dump 1uf filer1:nrst1a /vol/vol4`** filer1: **`dump 2uf urst1a /vol/vol2`**
Saturday	filer3: **`dump 0uf filer1:urst0a,urst0a,urst0a /vol/vol7`** filer2: **`dump 1uf filer1:nrst1a /vol/vol5`** filer1: **`dump 2uf urst1a /vol/vol3`**

Example of Backing Up the Entire Filer

This section provides a sample backup schedule for backing up a filer with seven volumes. The schedule uses the following assumptions:

- A low rate of change enables you to fit multiple incremental backups on one tape.
- Two incremental backups performed on each volume each week offer sufficient data protection due to a low rate of change.
- The level-0 backup takes 10 hours.

According to this schedule, the filer performs the following types of backups:

- The filer fully backs up each volume once a week.
- The filer performs a level-1 backup and a level-2 backup for each volume every week.
- The filer does not rewind the tape for each level-1 backup so that the subsequent level-2 backup can be written immediately after the level-1 backup on the tape.

Table 12-3 shows the sample backup schedule:

Table 12-3. Sample of Backing Up the Entire Filer

Day of week	dump commands
Sunday	**dump 0uf urst0a,urst0a,urst0a /vol/vol1** **dump 1uf nrst1a /vol/vol6** **dump 2uf urst1a /vol/vol4**
Monday	**dump 0uf urst0a,urst0a,urst0a /vol/vol2** **dump 1uf nrst1a /vol/vol7** **dump 2uf urst1a /vol/vol5**
Tuesday	**dump 0uf urst0a,urst0a,urst0a /vol/vol3** **dump 1uf nrst1a /vol/vol1** **dump 2uf urst1a /vol/vol6**
Wednesday	**dump 0uf urst0a,urst0a,urst0a /vol/vol4** **dump 1uf nrst1a /vol/vol2** **dump 2uf urst1a /vol/vol7**
Thursday	**dump 0uf urst0a,urst0a,urst0a /vol/vol5** **dump 1uf nrst1a /vol/vol3** **dump 2uf urst1a /vol/vol1**
Friday	**dump 0uf urst0a,urst0a,urst0a /vol/vol6** **dump 1uf nrst1a /vol/vol4** **dump 2uf urst1a /vol/vol2**
Saturday	**dump 0uf urst0a,urst0a,urst0a /vol/vol7** **dump 1uf nrst1a /vol/vol5** **dump 2uf urst1a /vol/vol3**

CHAPTER 13
Data Recovery

Introduction to Data Recovery

Why You Want to Restore Data From Tape

This section describes why you want to restore data to the filer from tape.

Files Were Deleted From Disk but Backed Up to Tape

For example, if you inadvertently delete a file and want to restore the file, you can recover the file from tape.

Files Are Corrupted

If some files are corrupted, you can restore the subtree containing the files.

No Disk Slots Are Available for Expansion

If the filer runs out of storage space, you can do the following tasks:

- Back up the entire filer
- Replace the current disks with disks of greater capacity
- Set up volumes on new disks
- Restore the filer from tapes

The Entire Filer Is Damaged and Unusable

If the entire filer is damaged and unusable, call Dell technical support to determine whether you can repair the filer.

If you can repair the filer but some files are deleted during the repair process, you need only restore the missing files. Otherwise, you need to reinitialize the disks and restore all files from tape.

CAUTION: Before initializing disks on your filer, call Dell technical support for instructions. Initializing disks destroys all data on your disks.

When You Do Not Recover Data From Tape

Recover a deleted file from tape only if you cannot recover the deleted file from any snapshot on the filer. If the file is in a snapshot, copying the file to the active file system is faster than recovering the file from tape.

Different Methods for Recovering Data

The filer supports these methods for recovering data that was backed up by the filer's `dump` command:

- Using the `restore` command on the filer to restore the file from a local or remote tape drive. If you use a remote tape drive, 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 Solaris `ufsrestore` command only if your filer runs NFS exclusively. If your filer runs the CIFS protocol, do not use the Solaris `ufsrestore` command. Doing so results in data loss.

What Data Cannot Be Recovered

The `restore` command cannot recover data that was backed up by commands issued on other systems. For example, do not try to use the filer's `restore` command to recover a tape file created by the Solaris `ufsdump` command.

UNIX FIle Permissions and Windows NT ACLs

A file recovered by the `restore` command has the same UNIX file permissions and Windows NT ACLs as it did when it was backed up. If you restore a file that has only UNIX file permissions to an NTFS qtree or volume, the file has no Windows NT ACLs. The filer uses only the UNIX file permissions on this file until you create Windows NT ACLs on it.

Scope of This Chapter

This chapter describes how to use the filer's `restore` command for data recovery. It does not describe how to use data recovery commands or programs on other systems. Refer to the documentation accompanying the other systems for information about their data recovery commands or programs.

The restore Command Syntax

The restore Command Syntax

The syntax for the `restore` command is as follows:

`restore [`*`function_key`*`] [`*`options`*`] [`*`arguments`*`] [`*`subtree`*`]`

Rules for Using the restore Command

Follow these rules when you enter the `restore` command:

- Specify no more than one function key.
- Specify multiple options without intervening spaces.
- Enter the arguments for each option in the order that you specify the options. Separate each argument from the next with a space.
- Place the subtree parameter after the last argument.

The restore Command Function Keys

Table 13-1 describes the function keys for the `restore` command.

Table 13-1. restore Command Function Keys

Key	Meaning
`r`	Rebuilds the file system or subtree. If you are applying incrementals, this must be the only option specified.
`R`	After the `restore` command is interrupted, this function key restarts data recovery from the last tape file used successfully by the command.
`t`	Lists all the file names on a tape. If you specify a path name, only the files in the path name are listed.
`x`	Extracts an individual file or subtree from the backup.

The restore Command options

Table 13-2 describes the options for the `restore` command.

Table 13-2. restore Command Options

Option	Argument	Meaning
`b`	*number*	Specifies the blocking factor. Use the same argument for the `b` option in the `dump` command.
`f`	*tape_file*	Specifies the name of the tape file. If you specify "-", `restore` reads from standard input.

Table 13-2. restore Command Options *(continued)*

Option	Argument	Meaning
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 backed up. If you created a backup before you installed multiple volumes on the filer, specify the path name into which the backup is restored. For example, if you backed up the */home* directory when the filer contained a single volume and you want to restore */home* to */vol/engineering/home*, specify */vol/engineering/home* as the target path name in the `restore` command. Otherwise, the */home* backup is restored to the */home* directory of the root volume.
s	*number*	Specifies the number of the file if multiple tape 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 restores files faster without the `v` option.
y		Specifies that `restore` not ask whether it should terminate 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 a read error, it terminates immediately.
A		Specifies that `restore` does not restore Windows NT ACLs.

Using the restore Command

Description

Use the `restore` command if you want to recover data that was backed up by the filer's `dump` command. After you recover the data, the files contain the same data as they did when you ran the `dump` command. You can use the `restore` command at any time.

Restrictions

This section describes the restrictions of the `restore` command.

The i Function Key of the Solaris Ufsrestore Command

The `restore` command is similar to the Solaris `ufsrestore` command, except that the `restore` command doesn't support the `i` function key of `ufsrestore`. This function key enables you to specify interactively individual files and directories to be restored from a tape. However, you can restore individual files and directories using the `x` argument of the filer `restore` command.

Incremental-Only Restores

If you want to restore all files in a backup using the `restore r` command, you must begin the restoration from a level 0 backup. You can, however, restore a specified file or directory from an incremental backup tape when you use the `restore x` command.

Parallel Restores

The filer supports up to three simultaneous `restore` commands.

Prerequisites

This section describes the requirements that you must meet for the `restore` command to be completed successfully.

- The space required for the `restore` command to be completed is about 100 MB more than the amount of data to be restored. The command terminates if it runs out of space. If you want to perform a full restore, the additional space should be in the directory that is the root of the backup. If you want to restore only some data in a backup, the additional space should be in the */etc/tmp* directory of the volume where data is to be restored.
- The `restore` command does not restore qtree information. Before you restore a volume containing qtrees, create the qtrees. Refer to "Performing a Full Restore of a Volume Containing qtrees" for the procedure for restoring qtrees.
- Before you perform a full restore, make sure that the directory into which you restore data does not contain the *restore_symboltable* file. The `restore` command uses the *restore_symboltable* file for incremental restores and for resuming an interrupted restore. If the *restore_symboltable* file exists in the directory because of an unsuccessful restore, remove the file before starting the full restore, or the full restore fails.

Where to Enter the restore Command

You can enter the `restore` command through the console or through `rsh`. Entering the `restore` command through `rsh` gives you these benefits:

- When the `restore` command is in progress, you can still use the console to manage the filer.
- You can start multiple `restore` commands through `rsh`.

- It is less likely that someone inadvertently terminates the `restore` command. If you enter a `restore` command on the console, it could be terminated by a Ctrl-C entered on a host connected to the filer using `telnet`.

However, if you enter the `restore` command on the console, you can read and respond to screen messages displayed by the command. For example, the command might prompt you for another tape to complete the recovery.

Steps

Follow these steps to restore files to the filer:

1. Place the tape containing the first tape file of the backup in the tape drive.
2. Enter the `restore` command as follows:

   `restore [`*`function_key`*`] [`*`options`*`] [`*`arguments`*`] [`*`subtree`*`]`

3. If prompted, insert the next tape in the backup.
4. Repeat Step 3 until the restore is complete.

Performing a Full Restore of a Volume Containing qtrees

Description

The `restore` command does not restore qtree information. For example, if you need to restore a volume or a part of a volume that contains qtrees, you must first create the qtrees into which data is to be restored.

Steps

Follow these steps to fully restore a volume containing qtrees:

1. Obtain the record of the qtree information as of the last volume backup.
2. Create all qtrees according to the record.
3. Set the security style of each qtree to the value that is the same as when the volume was backed up.
4. If you want to recover the entire volume with one `restore` command, go to Step 5. Using one `restore` command requires less time for recovering the entire volume than using multiple `restore` commands.

   If you want to use multiple `restore` command to recover all qtrees in the volume, go to Step 6. Using multiple `restore` commands enables you to quickly recover those qtrees that are needed the most.

5. Enter the `restore` command to recover the entire volume.

   Example: **`restore rfD rst0a /vol/vol0`**

6. Enter the `restore` command to restore each qtree in the volume.

   Examples: **`restore xfD rst0a /vol/vol0/NTusers /NTusers`**

   `restore xfD rst0a /vol/vol0/UNIXusers /UNIXusers`

Examples of the restore Command

Example of Restoring a Subtree

Perform the following steps to restore a subtree named */vol/vol0/home* from a local tape drive. The subtree was backed up as a subtree, not as a directory within a subtree.

1. Enter the `sysconfig -t` command on the filer to determine the name of the tape device.
2. Delete all files in the subtree before the restore.
3. Install the tape that contains the level 0 backup for the *home* subtree in a local tape drive.
4. Enter the following command if the subtree is backed up to only one tape:

   `restore rfD rst0a /vol/vol0/ho``me`

   You do not have to specify the `D` option and */vol/vol0/home* if the destination directory for the restore is the same as the directory that was backed up. This command restores the backup in `rst0a` to the */vol/vol0/home* directory, using the same block size as when the subtree was backed up to the tape.

   If the backup of the subtree is contained in multiple tape files, `restore` prompts you for the next tape file 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 backup.

   Example: If you have tapes containing level 1, level 2, and level 3 backups of */vol/vol0/home*, load the tape with the level 1 backup in the drive and repeat the `restore` command.

6. After the last incremental backup is restored, from a client, remove the *restore_symboltable* file in the directory that you just restored.

Example of 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 There Is One Backup for Each Volume

If you used one `dump` command to back up each volume, follow the procedure in "Example of Restoring a Subtree" to restore each volume. The only difference is that you use */vol/volume_name* as the directory to which the backup is restored.

If Each Volume Was Backed Up as Subtrees or qtrees

If you used separate `dump` commands to back up files, directories, and qtrees that make up each volume, restore each file, directory, and qtree. For example, if the root volume contains two directories, */vol/vol0/etc* and */vol/vol0/home*, and you used the `dump` commands to back them up separately, perform the following steps to restore the entire volume. In this example, the files are restored from a local tape drive.

1. If you are not restoring qtrees, go to Step 4.

   If you want to restore qtrees, go to Step 2.

2. Create directories on the filer using the `qtree` command that is used as the top of the subtrees to be restored.

   Examples:

   ```
   qtree etc

   qtree home
   ```

3. Set the security for each qtree created in Step 2.

4. Install the tape that contains the level 0 backup for the */vol/vol0/etc* subtree in the local tape drive.

   NOTE: The `restore` command doesn't support incremental-only restores. You must begin the restoration from a level 0 dump.

5. Enter the `sysconfig -t` command on the filer to determine the name of the tape device.

6. Enter the following command on the filer:

   `restore rfD rst0a /vol/vol0/etc`

   This command restores the backup in `rst0a` to the */vol/vol0/etc* directory.

   If the backup for the subtree is in multiple tape files, `restore` prompts you for the next tape volume when appropriate. See "Examples of Restoring From Multiple Tapes" for a sample screen display when more than one tape contains the dump volume.

7. After the restore is finished, remove the tape from the drive. Install another tape that contains the next lowest level of incremental dump.

   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 backup of the highest dump level, level 3 in this example, is restored.

8. From the client, remove the *restore_symboltable* file in the directory that you just restored.

9. Repeat Steps 4 through 8 for the */vol/vol0/home* subtree.

Examples of Restoring From Multiple Tapes

This section provides examples illustrating how to restore */vol/vol0* that was backed up to multiple tapes using the following command:

```
dump 0fB rst0a,rst1a 600 /vol/vol0
```

Restoring the Volume to a Directory From Multiple Tapes Using Two Tape Drives:

The following command restores the volume to the */vol/vol0/myexample* directory from two tapes that are in different tape drives:

```
restore rfD rst0a /vol/vol0/myexample
```

When the `restore` command prompts you for the next tape drive, enter the name of the tape drive, for example, `rst1a`.

Restoring a Volume to a Directory From Multiple Tapes Using One Tape Drive

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, for example, `rst0a`, when the filer prompts you for the name of the device for the second tape.

Example of 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.

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.

Example

The subtree containing the directory to be restored was backed up to two tapes:

```
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**.

Example of Listing Files

The following example lists the names of all files backed up to `rst0a`:

```
restore tf rst0a
```

The following example lists the names of all files in the */vol/vol0/home* directory on `rst0a` after you backed up the */vol/vol0* volume:

```
restore tf rst0a /home
```

Restarting the restore Command

Description

Restart the `restore` command if data recovery is interrupted for reasons such as a power outage or a Ctrl-C inadvertently entered by someone on the filer console. When you restart the `restore` command, the filer restarts data recovery from the

last tape file that was successfully used for restoring data. You do not have to start the command from the beginning of the backup.

Restrictions

The `restore` command has the following restrictions:

- You can restart a `restore` command only if the backup consists of multiple tape files. If the backup contains only one tape file, simply reenter the `restore` command to start the recovery from the beginning of the backup.
- You can restart a `restore` command only if the command is for a full restore. If the `restore` command is for extracting an individual file or subtree from a backup (that is, if you use the `x` function key), you cannot restart the command.

Steps

Enter the same `restore` command as the one that was interrupted with the following changes:

- Replace the `r` function key with the `R` function key to indicate that you want to restart the `restore` command.
- Use the appropriate tape file name for the `f` function key. The tape file should be the one used by the original `restore` command when it was interrupted.

Example

In this example, the following `restore` command was entered on the filer console for performing a full restore from three tape files in a tape stacker:

```
restore rfD urst0a,urst0a,rst0a /vol/vol0
```

When the `restore` command is recovering data from the second tape file, it is interrupted by a Ctrl-C entered on the filer console. Because the second tape file is on an unload/reload device, the filer closes the file and ejects the tape when the `restore` command is interrupted.

To restart the `restore` command, reload the second tape file and enter the following command.

```
restore RfD urst0a,rst0a /vol/vol0
```

The `restore` command restarts data recovery from the beginning of the second tape file of the backup.

How to Use a Filer Tape Drive to Restore Files to Another System

About This Section

This section describes the requirements for using commands such as `ufsrestore` on another system to restore data from a tape drive attached to the filer. It also describes how to specify the tape drive in the command for restoring data.

This section does not discuss the exact commands that you need to enter on the other system. Refer to the documentation for the other system to learn about its commands for restoring data.

Requirements

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 *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.

Format for Specifying Filer Tape Drive

In the commands for dumping and restoring, specify the filer tape drive in the following format:

filer:*device_name*

CHAPTER 14
Tape Device Management

Introduction to Tape Device Management

Why You Want to Manage a Tape Device

You need to manage a tape device when you back up data from the filer to tape or when you recover data from tape to the filer.

Scope of This Chapter

This chapter discusses the following topics:

- Displaying information about tape devices attached to the filer
- Positioning tapes

How the Filer Displays Information About Various Tape Drives

Introduction

This section describes how the filer displays tape drive information about different types of tape drives. You need the tape information for planning your backup.

This section does not describe how to display information about tape drives on remote hosts. The filer can only display information about local tape drives.

Qualified Tape Drives

Qualified tape drives are the tape drives that Dell tests with the filer. These tape drives are on the Dell price list.

Displaying Tape Device Information

Description

Before you use the `dump` command to back up to a tape device attached to the filer, verify that the filer detects the device. Also, verify the device name to be included in the `dump` command.

Step for Displaying Information About Qualified Tape Devices

To display information about qualified tape devices, enter the following command:

```
sysconfig -t
```

Steps for Displaying Nonqualified Tape Devices

To display information about nonqualified tape devices, perform the following steps:

1. If the filer has accessed the tape drive through the `dump` or `mt` command, go to Step 3. Otherwise, go to Step 2.
2. Enter the following command to access the tape device:

   ```
   mt -f device status
   ```

3. Enter the following command:

   ```
   sysconfig -t
   ```

Steps for Displaying Information About Tape Stackers

To display information about tape stackers, perform the following steps:

1. If the tape stacker's autoload setting is On, go to Step 2.

   If the tape stacker's autoload setting is Off, go to Step 4.
2. Turn off the autoload setting of the tape stacker.
3. Reboot the filer.
4. Enter the following command:

   ```
   sysconfig -m
   ```

Displaying Tape Device Information Along With Other Filer Information

You can use the `sysconfig -v` command to display the filer hardware information. Look for the information about the SCSI adapter to which the tape devices are attached.

Example of the sysconfig -t Command for a Qualified Tape Drive

The following example of the `sysconfig -t` command displays information about a qualified tape drive:

```
sysconfig -t
Tape drive (6.4)  Digital DLT7000
rst0l  -  rewind device,        format is: 81633 bpi 40 GB (w/comp)
nrst0l -  no rewind device,     format is: 81633 bpi 40 GB (w/comp)
urst0l -  unload/reload device, format is: 81633 bpi 40 GB (w/comp)
rst0m  -  rewind device,        format is: 85937 bpi 35 GB
nrst0m -  no rewind device,     format is: 85937 bpi 35 GB
urst0m -  unload/reload device, format is: 85937 bpi 35 GB
rst0h  -  rewind device,        format is: 85937 bpi 50 GB (w/comp)
nrst0h -  no rewind device,     format is: 85937 bpi 50 GB (w/comp)
urst0h -  unload/reload device, format is: 85937 bpi 50 GB (w/comp)
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)
```

Examples Of the sysconfig -t Command for a Nonqualified Tape Drive

The following example of the `sysconfig -t` command displays information about a nonqualified tape drive that the filer has not registered as a clone:

```
sysconfig -t
Tape drive (6.5) DLT9000 (Non-qualified tape drive)
```

The following example of the `sysconfig -t` command displays information about a nonqualified tape drive that the filer has registered as a clone:

```
sysconfig -t
Tape drive (6.5) DLT9000 emulates Digital DLT7000.
```

Example of the sysconfig -m Command

The following example of the `sysconfig -m` command displays information about a tape library attached to the filer:

```
sysconfig -m
Media changer (6.6)  BHTi     Quad 7
mc0 - media changer device
```

In this example, a tape library with SCSI ID 6 is attached to slot 6 of the filer.

Example of the sysconfig -v Command

The following example shows a tape stacker with SCSI ID 6 and a tape drive with SCSI ID 4 attached to slot 6 of the filer:

```
slot 6: SCSI Host Adapter 6 (QLogic ISP 1040B)
   Firmware Version 2.26          Clock Rate 60MHz.
   6: BHTi    Quad 7          1.41
   4: QUANTUM DLT7000         1B41
   In-Band Enclosure Services
```

Using the mt Command to Control Tape Devices

The mt Command Syntax

You can control tape devices with the `mt` command. 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 the end of data (end of media if the tape is full), as in the following example:

```
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:

```
mt -f nrst0a fsf 1
```

Rewinding a Tape

To rewind a tape, use the `rewind` command, as in the following example:

```
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:

```
mt -f nrst0a offline
```

Displaying Status Information

To display status information about the tape unit, use the `status` command, as in the following example:

```
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 15
Volume Copy Using the vol copy Command Set

About This Chapter

Overview of Volume Copy

The filer can copy data from one volume to another volume at a given time. The result of copying a volume is an off-line volume containing the same data as the source filer at the time you initiated the copy operation.

Introduction to the Filer's Commands for Copying Volumes

Purposes of the vol copy Command Set

The `vol copy` command set enables you to copy one volume to another. The commands in this command set control copying both data in the active file system and data in snapshots from one volume to another. The source and destination volumes of the copy can reside on the same filer or on different filers.

For more information about snapshots, see Chapter 9, "Snapshots."

When to Copy Volumes

Table 15-1 describes some situations where you might find copying volumes useful.

Table 15-1. vol copy Command Situations

Situation	Reasons for copying one volume to another
You want to copy data from one filer to another regularly to ensure high data availability.	After you copy the data, clients can switch to the destination filer in the following scenarios: • When you shut down the source filer for software or hardware upgrades. • If a network client process accidentally deletes a large number of files on the source filer, clients can continue to have access to the files when you are restoring the files to the source filer. • If the source filer is not available for reasons such as natural disasters, you can put the destination filer on-line to continue file service.
You want to migrate data from one filer to another.	The destination filer has more storage or is a model that supports newer technology, such as FC-AL disks.
You want to move a volume from one set of disks to another on the same filer.	Splitting a volume. **Example:** You can copy the *vol0* volume to the *vol1* volume and then delete duplicated files and directories in these volumes so that the original contents of *vol0* are split into two volumes.

Benefits of the vol copy Command Set

Although you can copy data on the filer using client programs such as `cpio` or using the filer's `dump` and `restore` commands, the `vol copy` command set offers the following benefits:

- When a `vol copy` command reads and writes data, the filer does not traverse directories on the filer. Data is copied block for block directly from the disks, which means that the filer can finish the copying faster than it could with other methods.
- Using a `vol copy` command, the filer preserves the snapshot data of the source volume. If, in the future, users might need to use snapshots that were taken before data was copied from one volume to another, for example, if users accidentally delete files and need to recover them, use a `vol copy` command for migrating data.

Requirements and Recommendation For Copying a Volume

Requirements for Copying a Volume

The filers involved in a volume copy operation must meet several requirements for data to be copied successfully. The following list is a brief description of these requirements. The rest of this section provides more detailed information about verifying whether the source and destination volumes meet these requirements.

- The source volume must be on-line and the destination volume must exist and be off-line.
- The destination volume must not be the root volume because the destination volume must be off-line when the filer executes the `vol copy` command, and a root volume must always be on-line.
- The capacity of the destination volume must be greater than or equal to the capacity of the source volume.
- The destination volume must not contain data that you want to preserve.
- The source and destination filers must have a trusting relationship with each other.
- The localhost interface must be specified as a trusted host if data is copied on the same filer.

Verifying the Status of Each Volume

The destination volume must exist before you enter the `vol copy start` command to start copying a volume. If the volume does not exist, the command does not create the volume and the command returns an error. See "Creating Volumes" in Chapter 3 for information about how to create a volume.

After you verify that the destination volume exists, check the status of the source volume and the destination volume.

Checking the Status of a Volume

The source volume must be on-line and the destination volume must be off-line. To verify whether a volume is on-line or off-line, use the following command syntax:

`vol status` *`volume_name`*

Changing the Status of a Volume

If you need to change the status of a volume, use one of the following command syntaxes:

- `vol offline` *`destination_volume`*
- `vol online` *`source_volume`*

NOTE: The `vol offline` *command takes effect only after you reboot the filer.*

Verifying the Size of Each Volume

The capacity of the destination volume must be greater than or equal to the capacity of the source volume, regardless of how many snapshots you try to copy. To verify the capacity of a volume, follow these steps:

1. Enter the following command syntax:

   `df /vol/`*`volume_name`*

   The command displays information about disk space in the active file system and in the snapshot for the specified volume.

2. Add the numbers in the kbytes column in the `df` output. The result is the number of kilobytes of data that the volume can hold.

Verifying the Contents of the Destination Volume

If the destination volume is not a new volume, make sure that the destination volume does not contain data that you might need in the future. This is because after the filer starts copying the source volume, it overwrites the entire destination volume. That is, all data in the active file system and in the snapshot of the destination volume is lost after the filer starts copying the data.

Verifying the Relationship Between Filers

If the source and destination volumes in a volume copy operation reside on two filers, the filers must have a trusting relationship with each other. That is, you must specify each filer as a trusted host in the */etc/hosts.equiv* file of the other filer. The */etc/hosts.equiv* file contains a list of host names, each of which is on a separate line.

Verifying Localhost as a Trusted Host

If you want to copy data between volumes on the same filer, you must specify localhost on the filer as a trusted host in the filer's */etc/hosts.equiv* file. Localhost is the interface through which the filer sends packets to itself.

If you have not already listed localhost as a trusted host, follow these steps to include localhost in */etc/hosts.equiv*:

1. Enter the loopback address, which is 127.0.0.1, in the */etc/hosts* file.

2. Type `localhost` on a separate line in the */etc/hosts.equiv* file.

Recommendation for Copying a Volume

To avoid performance problems when copying to a different filer, you might want to set up a private network for copying between the source and destination filers. This is because when a filer copies data between two volumes, it floods the network with

packets. Users of the filers involved in a volume copy operation might notice a degradation in response time during the copy.

Details About Copying One Volume to Another

Command Syntax for Copying One Volume to Another

To copy one volume to another, use the following command syntax:

```
vol copy start [ -S | -s snapshot ] source destination
```

Specifying the Snapshots to Copy

The `-S` and `-s` arguments specify the snapshots to copy. Table 15-2 describes the snapshots to copy and the resulting snapshots on the destination volume, depending on the argument.

Table 15-2. Command Syntax for Copying One Volume to Another

Argument used	Snapshots to copy from the source volume	Snapshots in the snapshot file system of the destination volume
None	The snapshot taken after you enter the `vol copy start` command	A snapshot named *snapshot_for_backup.n*, where *n* is a number starting at 0
`-S`	All snapshots in the snapshot file system of the source volume and the snapshot taken after you enter the `vol copy start` command	All snapshots in the source volume and *snapshot_for_backup.n*, where *n* is a number starting at 0
`-s` followed by the name of the snapshot	The specified snapshot	The specified snapshot

NOTE: The `vol copy start -S` *command does not copy any snapshots that are created when the copying is in progress. For example, if the copying lasts from 11:45 p.m. to 1:00 a.m. the next day, and the filer creates a snapshot named nightly.1 at midnight, the filer does not copy the nightly.1 snapshot.*

Specifying the Volumes Involved in the Copy

The *source* and *destination* arguments are the names of the volumes. If a volume is on a different filer, precede the volume name with the filer name and a colon. For examples illustrating how to specify volume names, see "Examples of the vol copy start Command."

Where to Enter the vol copy start Command

If the copying takes place between two filers, you can enter the command on either the source or destination filer. You cannot, however, enter the command on a third filer that does not contain the source or destination volume.

Examples of the vol copy start Command

Table 15-3 shows several examples of the `vol copy start` command.

Table 15-3. Examples of the vol copy start Command

If you want to...	Enter...
Copy all snapshots from the *vol0* volume to the *vol1* volume on the same filer	`vol copy start -S vol0 vol1`
Copy a nightly snapshot from the *vol0* volume to the *vol1* volume on the same filer	`vol copy start -s nightly.1 vol0 vol1`
Create a snapshot in the *vol0* volume to be copied to the *vol1* volume on the same filer	`vol copy start vol0 vol1`
Copy all snapshots from the *vol0* volume to the *vol1* volume on a different filer named filerA	`vol copy start -S vol0 filerA:vol1`

Results of the vol copy start Command

The `vol copy start` command generates volume copy operations and produces screen messages showing the progress of the operations.

Volume Copy Operations

Each `vol copy start` command generates two volume copy operations, as described in the following list:

- One operation is for reading data from the source volume. Screen messages displayed by a `vol copy` command refer to this operation as the `volcopy dump` operation.
- One operation is for writing data to the destination volume. Screen messages displayed by a `vol copy` command refer to this operation as the `volcopy restore` operation.

The filer assigns a volume copy operation number to each operation.

When to Use the Volume Copy Operation Number

You need the volume copy operation number if you want to stop a volume copy operation or change the volume copy operation speed.

For more information about obtaining the volume copy operation number, see "Checking the Status of a Volume Copy Operation."

Screen Messages From the vol copy Command

When the filer is copying a volume, the filer displays messages indicating the percentage of the volume copy operation completed and the number of minutes remaining. When the filer finishes copying all data, it displays the filer prompt.

Maximum Number of Simultaneous Volume Copy Operations

Each filer supports up to four simultaneous volume copy operations. Whether a filer can execute a `vol copy start` command depends on how many volume copy operations are already in progress on the filer or filers specified in the `vol copy start` command, as illustrated in the following examples.

Example

You can enter the following two `vol copy start` commands on a filer to copy volumes locally:

```
vol copy start vol0 vol1
vol copy start vol2 vol3
```

When these commands are in progress, you cannot enter additional `vol copy start` commands because four volume copy operations are already running on the filer. Two of the operations are for reading the *vol0* and *vol2* volumes, and two of the operations are for writing the *vol1* and *vol3* volumes.

Example

Suppose you enter the following three `vol copy start` commands on a filer named filerA to copy volumes to another filer named filerB.

```
vol copy start vol0 filerB:vol0
vol copy start vol1 filerB:vol1
vol copy start vol2 filerB:vol2
```

When these commands are in progress, filerA runs three volume copy operations to read the volumes, and filerB runs three volume copy operations to write the volumes.

You can enter an additional `vol copy start` command to copy between filerA and filerB because the command adds one more volume copy operation to each filer.

However, you cannot enter an additional `vol copy start` command to copy volumes locally on either filerA or filerB. This is because the additional command would create two volume copy operations, one for reading and one for writing, on the filer

that performs the local copying. The filer cannot support these two additional volume copy operations because three operations are already in progress.

Possible Errors

If your filer does not meet a requirement described in "Requirements and Recommendation for Copying a Volume," the `vol copy start` command generates one or more error messages. Table 15-4 explains the meanings of the possible error messages.

Table 15-4. vol copy start Command Error Messages

Error message	Meaning

`Permission denied.`
 	
`VOLCOPY: Could not connect to filer 127.0.0.1.`
	The source filer does not have permission to copy to the destination filer.
	
	Action: Make sure that the filers have a trusting relationship with each other. If both the source volume and destination volume are on the same filer, remember to include localhost in the */etc/hosts.equiv* file.

`VOLCOPY: volcopy restore: volume is online, aborting`	The destination volume is on-line. **Action:** Take the destination volume off-line and reboot the filer that contains the destination volume.
`VOLCOPY: volcopy restore: volume is too small, aborting`	The destination volume is smaller than the source volume. **Action:** Add more disk space to the destination volume or choose another destination volume of sufficient capacity.

Management of a Volume Copy Operation When it Is in Progress

Checking the Status of a Volume Copy Operation

You can use the following command syntax to check the status of one or more volume copy operations:

```
vol copy status [ operation_number ]
```

This command displays the status for a specified volume copy operation. If you do not specify the operation number, the command displays the status of all volume copy operations in progress. In the command output, the operations are differentiated from one another with unique volume copy operation numbers, ranging from 0 to 3. For more information about volume copy operation numbers, see "Results of the vol copy start Command."

Where to Enter the vol copy status Command

If you start a volume copy operation from the filer's console, you can enter the `vol copy status` command only through `rsh` when the copy operation is in progress. This is because you do not have access to the filer prompt on the console when the filer is copying the volume.

If data is being copied between two filers, you can enter this command through an `rsh` connection to either filer. The operation numbers displayed on the source filer and the destination filer are different because the reading and the writing are considered two different operations.

Example of a vol copy status Command

The following example illustrates a `vol copy start` command that copies the *vol0* volume to the *vol1* volume on the same filer:

```
vol copy start -S vol0 vol1

Copy Volume: vol0 on machine 127.0.0.1 to Volume: vol1
Reading the dump stream
VOLCOPY: Starting on volume 1.
This dump contains 257 blocks
10:04 pm : volcopy restore 1 : begun.
10:04 pm : volcopy restore 1 : 5 % done. Estimate 3 minutes remaining.
.
.
.
10:04 pm : volcopy restore 1 : 95% done. Estimate 1 minutes remaining.
```

Before the filer prompt is displayed again, you can use the `vol copy status` command on a trusted host of the filer, as shown in the following example:

```
rsh filer vol copy status

10:04 pm : volcopy dump 0 : 99 % done. Estimate 1 minutes remaining.
10:04 pm : volcopy restore 1 : 99 % done. Estimate 1 minutes remaining.
No operation 2 in progress.
No operation 3 in progress.
```

In this example, volume copy operation 0, shown as `volcopy dump 0` in the display, is for reading the data from the *vol0* volume; volume copy operation 1, shown as `volcopy restore 1` in the display, is for writing the data to the *vol1* volume.

Aborting a Volume Copy Operation

To stop a volume copy operation, use the following command syntax:

`vol copy abort [ operation_number ]`

The *operation_number* parameter specifies the volume copy operation to be aborted. You can obtain the operation number from the `vol copy status` output.

CAUTION: If you do not specify an operation number in the command, the filer aborts *all* volume copy operations. It does not display a help string for the `vol copy abort` command.

If data is being copied between two filers, you can execute this command on either filer.

If you start the volume copying operation from the filer console, you can enter the `vol copy abort` command only through `rsh` because you do not have access to the filer prompt on the console during the copying.

CAUTION: An incomplete volume copy operation leaves unusable data in the destination volume.

Controlling the Speed of a Volume Copy Operation

You can control the speed of a volume copy operation in two ways.

- Before you enter the `vol copy start` command, use the following command syntax to set the default speed for all volume copy operations:

  `options vol.copy.throttle value`

  `The value variable specifies the speed, which ranges from 10 (full speed) to 1 (one-tenth of full speed).`
- If a volume copy operation is in progress, use the following command syntax to set the speed of a specific operation:

  `vol copy throttle [ operation_number ] value`
  In this case, because a volume copy operation is in progress and you do not have access to the filer prompt, you must enter the `vol copy throttle` command through `rsh`.

  The *operation_number* parameter specifies the volume copy operation whose speed you want to adjust. If you do not specify an operation number, the command applies to all volume copy operations that are in progress. The *value* variable specifies the speed, which ranges from 10 (full speed) to 1 (one-tenth of full speed).

NOTE: The speed for reading data from the source volume and the speed for writing data to the destination volume can be different. In this case, the smaller of the two values determines the time required for the filer to finish copying the data.

Displaying the Default Speed for Copying a Volume

Before starting the `vol copy start` command, you can verify the default speed for all volume copy operations using the following command:

```
options vol.copy.throttle
```

It displays the value (1 through 10) to be used by all volume copy operations. The value of the `vol.copy.throttle` option was set at 10 at the factory.

Example of Controlling the Speed of Copying a Volume

The following example illustrates changing the speed of all volume copy operations in progress to one-tenth of full speed through `rsh`:

```
rsh filer vol copy throttle 1

volcopy operation 0: Throttle adjusted from 100% to 10%.
volcopy operation 1: Throttle adjusted from 100% to 10%.
```

CHAPTER 16
Data Replication Using SnapMirror

About This Chapter

Overview of SnapMirror

The SnapMirror feature replicates data from one volume (the source) to another volume (the mirror) and periodically updates the mirror to reflect incremental changes on the source. The result of this process is an on-line, read-only volume that contains the same data as the source volume at the time of the most recent update. SnapMirror requires a license code.

Purposes of SnapMirror

Why You Want to Replicate a Volume

You want to replicate a volume if you are in any of the situations described in "When to Copy Volumes." Because SnapMirror offers additional advantages, there are other situations where you want to replicate a volume, as described in Table 16-1.

Table 16-1. Replicating a Volume Situation

Situation	How data replication helps
Remote access to data: Users who need read access to a volume are distributed over a large geographical area.	You can replicate the source volume on other filers that are geographically closer to the users. Users accessing a local filer can read the data faster than they could if they connected to a distant filer.
Load balancing: A large number of users need only read access to a volume.	Replicating a volume on multiple filers enables you to distribute the load.

Table 16-1. Replicating a Volume Situation *(continued)*

Situation	How data replication helps
Backup: You need to reserve all processing and networking resources on a filer for serving NFS and CIFS requests.	After replicating data on the source filer, you can back up the data in the mirror to tape. This means the source filer does not have to allocate resources for performing backups.
Data migration: You want to migrate data from one filer to another without interrupting network service.	After replicating the source filer, you can export or share the source filer with read-only permissions so that network clients cannot make additional changes to the source filer. Because the source and destination filers contain identical data, you can change the status of each mirror on the destination filer to a regular volume. For more information about how to convert a mirror to a regular volume, refer to "Setting Volume Options" in Chapter 3.
Disaster recovery: You want to provide immediate access to data after some disaster has caused a volume or a filer to be unavailable.	You can change the status of the mirror to a regular volume. Users can immediately have access to the same data as that on the source volume. The `snapmirrored` volume option controls whether a volume is a mirror or a regular volume. For more information about volume options, refer to "Setting Volume Options" in Chapter 3.

How SnapMirror Works

Command and Configuration File for Controlling SnapMirror

To use SnapMirror for replicating a volume, use the vol snapmirror command set and two configuration files: /etc/snapmirror.conf and /etc/snapmirror.allow. The command set is described in "Replicating a Volume." The configuration files are described in "The /etc/snapmirror.allow File" and "The /etc/snapmirror.conf File."

How the Filer Creates a Baseline Version of the Mirror

To use SnapMirror, you create an off-line volume to be used as the destination for data replication. The destination volume is referred to as the "mirror." The mirror can reside on the same filer as the source volume or on another filer.

To replicate data for the first time, the filer transfers all data in all snapshots in the source volume to the mirror. After the filer finishes transferring the data, it brings the mirror on-line. This version of the mirror is the base line for future incremental changes.

How the Filer Updates the Mirror

To make incremental changes on the mirror, the filer takes regular snapshots on the source volume according to the schedule specified in the */etc/snapmirror.conf* file. By comparing the current snapshot with the previous snapshot, the filer determines what changes it needs to make to synchronize the data in the source volume and the data in the mirror. For example, if a file is deleted from the source volume, SnapMirror deletes the corresponding file in the mirror the next time it updates the mirror.

For more information about how SnapMirror uses snapshots to transfer data from the source volume to the mirror, refer to "Snapshots Created During Data Replication."

Number of Volume Copy Operations SnapMirror Generates

When SnapMirror transfers data from one volume to another, it generates two volume copy operations in the same way as the `vol copy` command. Refer to "Volume Copy Operations" in Chapter 18 for the meaning of a volume copy operation.

What Happens After You Replicate a Volume

After you replicate a volume, the active file system and all snapshots in the source volume are available on the mirror. As with any volume, you can export the mirror for NFS mounting or add a share corresponding to this volume for CIFS sharing.

SnapMirror makes all changes to the mirror at the same time. If you have an open file in the mirror while the filer is updating the mirror, you do not see the changes immediately. After the mirror update is finished, if you open the file, you see the changes. This is similar to the situation where another user changes a file in a regular volume when you are reading the file. You can see the changes the next time you open the file.

Differences Between a Mirror and a Regular Volume

The following list describes the differences between a mirror and a regular volume:

- A mirror has snapmirrored status, which means that it contains read-only data and network clients cannot write data to the mirror.
- The filer does not create automatic snapshots on the mirror based on the snapshot schedule.
- You cannot use the `qtree create` command to create qtrees on a mirror. However, if qtrees exist in the source volume, the filer mirrors the qtrees to the mirror.
- You cannot enable quotas on a mirror.

Snapshots Created During Data Replication

Naming Conventions for Snapshots Used by SnapMirror

SnapMirror creates snapshots on the source volume, which are copied to the mirror. The snapshot name is in the following format:

filer_volume.number

filer is the host name of the destination filer.

volume is the name of the destination volume (the mirror).

number is the number for the snapshot starting at 1.

SnapMirror automatically deletes old snapshots that are no longer necessary for data replication.

Example

The following example describes the snapshots that are created on the source volume and copied to the mirror.

In this example, data is replicated from */vol/vol1*of filerA to */vol/vol2* of filerB.

To create a baseline version of a mirror, filerA creates a snapshot file named */vol/vol1/filerB_vol2.1* on filerA. All snapshots in */vol/vol1* of filerA, including */vol/vol1/filerB_vol2.1*, are transferred to */vol/vol2* of filerB.

For example, the `snap list` command on filerB generates the following display after */vol/vol1/filerB_vol2.1* is transferred from filerA to filerB.

```
Volume vol2
working.....

%/used    %/total  date          name
--------  --------  ------------  --------
0% ( 0%) 0% ( 0%)  Nov 17 10:50  filerB_vol2.1
1% ( 0%) 0% ( 0%)  Nov 17 10:00  hourly.0
1% ( 0%) 0% ( 0%)  Nov 17 00:00  nightly.0
1% ( 0%) 0% ( 0%)  Nov 15 16:00  hourly.1
1% ( 0%) 1% ( 0%)  Nov 15 15:00  hourly.2
2% ( 0%) 1% ( 0%)  Nov 15 14:00  hourly.3
2% ( 0%) 1% ( 0%)  Nov 15 13:00  hourly.4
2% ( 0%) 1% ( 0%)  Nov 15 12:00  hourly.5
```

When it is time to update the mirror, another snapshot is created on filerA. The `snap list` command on filerA generates the following display after */vol/vol1/filerB_vol2.2* is created on filerA:

```
Volume vol1
working....
%/used   %/total     date          name
-------- ----------  ------------  --------
0% ( 0%) 0% ( 0%)    Nov 17 10:52  filerB_vol2.2 (busy)
0% ( 0%) 0% ( 0%)    Nov 17 10:51  filerB_vol2.1
1% ( 0%) 0% ( 0%)    Nov 17 10:00  hourly.0
1% ( 0%) 0% ( 0%)    Nov 17 00:00  nightly.0
1% ( 0%) 0% ( 0%)    Nov 15 16:00  hourly.1
1% ( 0%) 1% ( 0%)    Nov 15 15:00  hourly.2
```

After *filerB_vol2.2* is transferred from filerA to filerB, both the *filerB_vol2.1* and *filerB_vol2.2* snapshots exist on filerB.

On filerA, however, *filerB_vol2.1* is no longer needed and is deleted; only *filerB_vol2.2* remains.

Consequences of Deleting a Required Snapshot

The filer fails to replicate data if it cannot find the required snapshots in the source and destination volumes. This section provides an example illustrating what happens if you inadvertently delete a snapshot required for data replication.

In this example, data is replicated from */vol/vol1* of filerA to */vol/vol2* of filerB. Suppose you use the `snap delete` command on filerA to delete the ***`filerB_vol2.2`*** snapshot. When filerB tried to update the mirror according to the schedule in the *snapmirror.conf* file, it could not find the snapshot to determine what incremental changes are required in the mirror.

As a result, filerB displays the following error message.

```
The source filer requires a complete transfer. The destination volume
is online and must be offline for a complete transfer.
```

FilerA displays the following error message:

```
Destination filerB could not accept a complete transfer.
```

These messages mean that filerB tries to create a baseline version of the mirror, but doing so requires that the mirror on filerB be taken off-line.

To proceed, use the `vol offline` command to take the mirror off-line and reboot the filer. At the next scheduled mirror update, the filer creates a baseline version of the mirror.

How SnapMirror Works With Quotas

Quotas on the Mirror

Quotas are always disabled on a mirror, regardless of whether quotas are enabled on the source volume. If you try to enable quotas on a mirror, the filer generates an error message.

How to Apply the Same Quota Restrictions on the Former Mirror

After you convert a mirror to a regular volume, you can enable quotas on it.

However, if the former mirror and the source volume reside on different filers, they might be under the effect of different quota restrictions because there is one */etc/quotas* file for each filer. If you want the same quota restrictions to be applied on both volumes, make sure that the filer on which the former mirror resides has a */etc/quotas* file containing all entries from the */etc/quotas* file used by the source volume.

Refer to "Converting a Mirror to a Regular Volume" for information about changing the status of a mirror.

How SnapMirror Works With the Dump Command

How to Back Up Data in the Mirror

If you want to back up data from a mirror, the data must be in an existing snapshot, not the active file system, on the mirror. This is because before the `dump` command writes data from an active file system to tape, the filer must create a snapshot in the volume containing the data. The filer cannot create snapshots on the mirror, which is a read-only volume, to be used by the `dump` command.

You can back up any snapshot displayed by the `snap list` command on the mirror. You can also create a snapshot on the source volume, replicate the snapshot to the mirror, and use the `dump` command to back up this snapshot from the mirror to tape.

Effect of the Dump Command on the Mirror Update Schedule

Running the `dump` command on a mirror might change the mirror update schedule in the following ways:

- If the mirror is scheduled to be updated when a `dump` command is in progress on the mirror, the filer delays replicating data from the source volume until the `dump` command is finished.

- If you start the `dump` command when the filer is replicating data to the mirror, the filer stops the replication and does not update the mirror. It restarts replication after the `dump` command is finished.

Delaying the mirror update is necessary because when the `dump` command backs up data from a snapshot, the snapshot must exist until the command is completed. During a mirror update, the filer deletes existing snapshots on the mirror and copies new ones from the source. This process causes the snapshot being backed up to disappear and the `dump` command to terminate.

The /etc/snapmirror.allow File

Purpose of the snapmirror.allow File

The */etc/snapmirror.allow* file on the source filer specifies the host names of filers that are allowed to replicate data from the source filer.

The filer is not shipped with a default */etc/snapmirror.allow* file. You must use a text editor to create the file if you want to use SnapMirror.

When You Can Modify the snapmirror.allow File

You can modify the */etc/snapmirror.allow* file at any time.

Format of the snapmirror.allow File

Each entry in the */etc/snapmirror.allow* file contains the host name of the filer that can replicate data from the source filer. Type each entry on a separate line.

Example

If you want to replicate volumes locally on filerA, enter this line in the */etc/snapmirror.allow* file on filerA:

```
filerA
```

Example

If you want to replicate volumes from filerA to filerB, enter this line in the */etc/snapmirror.allow* file on filerA:

```
filerB
```

The /etc/snapmirror.conf File

Purpose of the snapmirror.conf File

The */etc/snapmirror.conf* file resides on the destination filer. It controls where data is copied and how often a mirror is updated.

The filer is not shipped with a default */etc/snapmirror.conf* file. You must use a text editor to create the file if you want to use SnapMirror.

When You Can Modify the snapmirror.conf File

You can modify the */etc/snapmirror.conf* line at any time.

Format of the snapmirror.conf File

Each entry of the */etc/snapmirror.conf* file is in the following format:

source_filer:source_vol destination_filer:destination_vol argument schedule

Meaning of Each Field in asnapmirror.conf Entry

The following list describes the meaning of each field in a *snapmirror.conf* entry:

- *source_filer* is the host name of the filer from which data is replicated.
- *source_vol* is the source volume name. For example, for the *vol1* volume, type **`vol1`** Do not type the full path name (*/vol/vol1*).
- *destination_filer* is the host name of the filer to which data is replicated.
- *destination_vol* is the destination volume (mirror) name. For example, for the *vol1* volume, type **`vol1`** Do not type the full path name (*/vol/vol1*).
- *argument* is kbs (kilobytes per second), the maximum speed at which data is transferred. Enter a value greater than or equal to 11. By default, the filer transfers the data as fast as it can. Enter a dash (-) to indicate that you want to use the default value for *argument*.

NOTE: The actual data transfer speed might be limited by factors such as network bandwidth. For example, if you specify a large value such as 1,000,000, the filer might still transfer the data at 2,000 kilobytes per second.

- *schedule* is the schedule used by the destination filer for updating the mirror.

You must not leave any field in a *snapmirror.conf* entry blank. If you do not want to specify the maximum speed for data transfer, specify a dash as the *argument*.

Rules for Specifying the Update Schedule

The schedule in each */etc/snapmirror.conf* entry contains four fields:

- minute
- hour
- day of month
- day of week

The fields are separated from each other by a space. If a field contains more than one value, the values are separated from each other by a comma. A field containing an asterisk (*) means that the field is irrelevant. If you specify an asterisk in each field of the schedule, the filer updates the mirror every minute.

The update schedule is mandatory. The filer generates an error message if a */etc/snapmirror.conf* entry does not contain a schedule.

Example

```
filerA:vol1 filerB:vol2 kbs=2000 45 10,11,12,13,14,15,16 * 1,2,3,4,5
```

In this example, */vol/vol1* on filerA is replicated to */vol/vol2* on filerB. Data is replicated at a maximum rate up to 2,000 kilobytes per second. FilerB updates */vol/vol2* at 10:45 a.m., 11:45 a.m., 12:45 p.m., 1:45 p.m., 2:45 p.m., 3:45 p.m., and 4:45 p.m., Monday through Friday. The asterisk in this example means that the mirror update schedule is not affected by the day of month.

When Changes to snapmirror.conf Take Effect

If SnapMirror is enabled, the changes take effect within two minutes. If SnapMirror is disabled, the changes take effect immediately after you enter the `vol snapmirror on` command to enable the feature.

Recommendation

You can create a single */etc/snapmirror.conf* file for your site and copy it to all the filers that use SnapMirror. The */etc/snapmirror.conf* file can contain entries pertaining to other filers. For example, the */etc/snapmirror.conf* file on filerB can contain an entry for replicating a volume from filerC to filerD. When filerB reads the */etc/snapmirror.conf* file, it simply ignores this entry.

Replicating a Volume

Description

Replicate a volume if you are in any situation described in "Purposes of SnapMirror." You can replicate a volume at any time. After you replicate a volume, you have a mirror that contains the same data as the source volume.

Prerequisites

The following prerequisites must be met before you can replicate a volume:

- You must purchase the SnapMirror license. If the source volume and the mirror are on different filers, you must purchase a license and enter the SnapMirror license code for each filer. Refer to "Enabling Services" in Chapter 2 for information about how to enter a license code.
- You must create an off-line volume to be used as the mirror. SnapMirror does not automatically create a volume. Refer to "Creating Volumes" in Chapter 3 for information about how to create a volume; refer to "Making a Volume Inactive" in Chapter 3 for information about how to take a volume off-line.
- The source volume must be on-line. Refer to "Reactivating an Off-line Volume" in Chapter 3 for information about how to put a volume on-line.
- The source volume must not be a mirror.
- The capacity of the mirror must be greater than or equal to the capacity of the source volume. The configuration of the volumes, however, can be different. For example, the RAID group size can be different for the two volumes. Refer to "Adding Disks to a Volume" in Chapter 3 for information about how to add disks to a volume.
- The mirror must not be the root volume.

Restrictions

Each filer supports up to four simultaneous volume copy operations. For information about volume copy operations, refer to "Number of Volume Copy Operations SnapMirror Generates."

Cautions

- SnapMirror creates snapshots in the source volume, which are copied to the mirror. Do not delete these snapshots because incremental changes to the mirror depend on the snapshots. If the filer cannot find the required snapshot, it cannot perform incremental changes to the mirror.

  Refer to "Snapshots Created During Data Replication" for an example about how snapshots are created on both the source volume and the mirror.

- The mirror must have snapmirrored status. Do not disable the snapmirrored status using the `volume options` command. Otherwise, the mirror becomes a regular, writable volume. Disable the snapmirrored status only when you no longer need to mirror incremental changes from the source volume.

Recommendations

Follow these recommendations to minimize confusion or to replicate data efficiently:

- When specifying the mirror update schedule in the */etc/snapmirror.conf* file, stagger the update times instead of starting multiple mirror updates at the same

time. If the filer does not have enough resources to perform all scheduled mirror updates, it postpones some updates until it has enough resources to do so. As a result, the filer might need to perform subsequent updates at times that are different from those you specify in the */etc/snapmirror.conf* file.

- During data replication, the filer copies data from all snapshots from the source volume. Therefore, the filer must preserve all snapshots when data replication is in progress, even though the snapshot schedule might call for the deletion of some snapshots. If you want the filer to delete snapshots at the exact time specified by the `snap sched` command, schedule the mirror updates to be different from the snapshot deletion times.
- The filer transfers data faster if the source volume and the mirror have the following characteristics:
  - They contain disks of the same size.
  - They contain RAID groups of the same size.
  - They contain the same number of RAID groups.
- If the source volume has quotas enabled and you want to apply the same quota restrictions to the mirror after converting the mirror to a regular volume, make sure that the filer on which the mirror resides contains a copy of the */etc/quotas* file from the source filer. Whenever you make a change to the */etc/quotas* file on the source filer, make the same change to the copy of the */etc/quotas* file on the destination filer.

Steps

Follow these steps to replicate a volume:

1. Add the host name of the destination filer to the */etc/snapmirror.allow* file on the source filer.
2. Edit the */etc/snapmirror.conf* file on the destination filer to specify the volume to be replicated and the schedule at which the mirror is updated.
3. Enter the `vol snapmirror on` command on both the source filer and destination filer to enable SnapMirror.

   Result: The filer reads the */etc/snapmirror.conf* file. If the filer is a destination filer specified in the */etc/snapmirror.conf* file, it establishes a connection with the source filer at the time of the scheduled mirror update.

   If a baseline version of the mirror does not exist, the filer takes a snapshot of the source volume at the time of the scheduled mirror update and transfers all data in the snapshot from the source volume to the mirror.

   The filer does not display any messages if it can transfer the data successfully. You can check the data replication status following the instructions in "Checking Data Replication Status."

   The filer makes subsequent updates to the mirror according to the schedule specified in the */etc/snapmirror.conf* file.

NOTE: The `vol snapmirror on` *command does not persist across filer reboots. Put the command in the /etc/rc file if you want the command to remain in effect after the filer is rebooted. If the /etc/rc file does not contain a* `vol snapmirror` *command, data replication is disabled.*

4. If the source volume and the mirror reside on different filers, and if the source volume has quotas enabled, copy the */etc/quotas* file from the source filer to a file on the destination filer. Whenever you make a change to the */etc/quotas* file on the source filer, make the same change to the copy on the destination filer so that the destination filer always contains a record of all the quota entries used by the source filer.

Disabling Data Replication for the Entire Filer

Description

You can disable data replication for the entire filer if you decide that data replication is no longer necessary. You can disable the feature at any time, even when replication is underway. The mirror remains unchanged after you disable replication.

Steps

Follow these steps to disable volume replication for the entire filer:

1. Enter the `vol snapmirror off` command on the destination filer to disable SnapMirror.

   Result: If the filer is currently transferring data from one volume to another, the transfer stops immediately. The destination volume remains the same as before the transfer. The snapshot taken in the source volume for the data transfer remains, but is deleted and replaced by a new snapshot the next time the mirror is updated.

   The filer stops reading the */etc/snapmirror.conf* file every minute.

   Entering the `vol snapmirror off` command on the destination filer does not affect SnapMirror on the source filer. Other filers can continue to replicate data from the source filer.

2. If the `vol snapmirror on` command is in the */etc/rc* file, remove the command.

NOTE: There is no need to enter the `vol snapmirror off` *command in the /etc/rc file. By default, data replication is disabled.*

Resuming Data Replication for the Entire Filer

Description

After you disable data replication, you can resume it at any time. When data replication resumes, the filer copies the data from the source volume to the mirror that has changed since the last update.

Prerequisites

In addition to the requirements described in "Prerequisites" on page 10, you must meet these requirements before you can resume data replication:

- The mirror must have snapmirrored status. If you converted the mirror to a regular volume, the filer cannot resume data replicating on the regular volume. If you want to use the regular volume as a mirror again, follow the procedure in "Replicating a Volume" so that the filer can create a baseline version of the mirror. Do not try to assign the snapmirrored status to the volume, because you cannot set the `snapmirrored` volume option to On.
- If the mirror never lost the snapmirrored status but you took the mirror off-line, you must put it back on-line.

Step

To resume data replication for the entire filer, enter the `vol snapmirror on` command on the destination filer to enable SnapMirror.

Result: The filer reads the */etc/snapmirror.conf* file to determine whether it needs to create a baseline version of a mirror or to update a mirror.

Disabling Data Replication for One Volume

Description

You can stop data replication for a particular volume if you decide that there is no need to update the mirror. For example, you might want to change the mirror to a regular volume.

You can disable data replication at any time, even when data transfer is underway. The destination volume remains the same as before the transfer. The snapshot taken in the source volume for the data transfer remains, but is deleted and replaced by a new snapshot the next time the mirror is updated.

Steps to Disable Data Replication for One Volume

Follow these steps to disable data replication for one volume. These steps disable volume replication until you reenable it.

1. Comment out the entry in the */etc/snapmirror.conf* file by preceding the entry with a pound sign (#).
2. Enter the `vol snapmirror on` command to make the filer reread the */etc/snapmirror.conf* file.

   Result: If the filer is currently transferring data from that volume, the transfer stops immediately.

Steps to Disable Data Replication While Data Transfer Is in Progress

Follow these steps to disable data replication while data transfer is in progress. These steps stop the replication temporarily; at the time of the next scheduled update, data replication starts up again.

1. Enter the `vol copy status` command to display the operation numbers of the volume copy.

   Result: Because SnapMirror uses the `vol copy` command to transfer data, the `vol copy status` command displays all volume copy operations, including the one for replicating a volume.
2. Enter the `vol copy abort` command to terminate the volume copy operation.

   Result: The filer stops copying data and displays messages similar to these:

   ```
   image operation 0 is now being aborted

   image operation 1 is now being aborted
   ```

Checking Data Replication Status

Description

If you want to know how much data has been copied to the mirror or whether a filer is using resources for replicating data, you can use the `vol snapmirror status` command at any time to check the status of data replication.

Prerequisite

SnapMirror must be enabled before you can check the status of data replication. Otherwise, the following error message is displayed:

```
Snapmirror is turned off.
```

Step

To check the status of data replication, enter the `vol snapmirror status` command.

Result: The filer displays a message showing whether a transfer is in progress or how much data replication has been completed.

Examples

The following examples describe how the `vol snapmirror status` command displays the status of data replicating.

When No Data Replication Is in Progress

The following display shows that currently no data is being copied from the *vol0* volume to the *vol1* volume on filerA:

```
Source        Dest            Status
filerA:vol0  filerA:vol1      Idle
```

When Data Replicating Is in Progress

The following display shows that the filer has just begun transferring data from the *vol0* volume to the *vol1* volume on filerA:

```
Source        Dest            Status
filerA:vol0  filerA:vol1    Transferring
```

The following display shows that the filer finished transferring 26% of the data from the *vol0* volume to the *vol1* volume on filerA:

```
Source        Dest            Status
filerA:vol0  filerA:vol1    Transferring (26% complete)
```

Converting a Mirror to a Regular Volume

Description

If you use SnapMirror for data migration, after you synchronize the data between the source volume and the mirror, convert the mirror to a regular volume. If you use SnapMirror for disaster recovery, convert the mirror to a regular volume after the source volume becomes unavailable. After the conversion, you can export the volume or create a share for the volume so that network users can write to it in the same way as they did to the source volume.

Prerequisite

You must meet this prerequisite if the source volume and the mirror reside on different filers and you want the same quota restrictions to be applied after converting the mirror to a regular volume:

The destination filer must have a */etc/quotas* file that includes all entries from the */etc/quotas* file used by the source filer. If SnapMirror is used for data migration, you can copy the */etc/quotas* entries from the source filer to the */etc/quotas* file of the destination filer before you convert the mirror to a regular volume. However, if SnapMirror is used for disaster recovery, on the destination filer, keep a copy of all */etc/quotas* entries used by the source filer at all times so that you can apply the entries to the destination volume when the source filer becomes unavailable.

Steps

Follow these steps to convert a mirror to a regular volume:

1. If you want to enable quotas after converting the mirror to a regular volume, go to Step 2. Otherwise, go to Step 4.
2. Edit the */etc/quotas* file on the destination filer so that after the conversion, the former mirror has the same quota restrictions as the source volume.

   If the source volume uses per-volume quotas, replace the source volume name with the mirror name in the quota entries.
3. Enter the following command to convert the mirror to a regular volume:

   ```
   vol options volume_name snapmirrored off
   ```

   If you want to enable quotas on the former mirror, go to Step 4. Otherwise, the procedure is complete.
4. Enter the following command to enable quotas on the former mirror:

   ```
   quota on volume_name
   ```

Differences Between the vol copy Command and SnapMirror

Differences

Table 16-2 describes the differences between the `vol copy` command and SnapMirror.

Table 16-2. Differences in vol copy Command and SnapMirror

The vol copy command	SnapMirror
It is a standard Data ONTAP 5.3 feature that requires no license codes.	It requires a license code.
The result of copying a volume is an off-line volume.	The result of replicating a volume is a mirror, which is an on-line, read-only volume.
It does not copy incremental changes from the source volume.	It periodically copies incremental changes from the source volume.
It requires that you enter the destination filer name in the */etc/hosts.equiv* file.	It requires that you enter the destination filer name in the */etc/snapmirror.allow* file.
It does not involve a configuration file that controls how a volume is copied.	It requires that you specify an entry in the */etc/snapmirror.conf* file to control the volume to be replicated, the data transfer rate, and the mirror update schedule.

CHAPTER 17
System Information and Performance

Displaying the Data ONTAP Version

How to Display 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:

```
version

NetApp Release 5.3: Fri May 12 03:06:00 PDT 1998
```

Displaying Filer Configuration Information

Use the sysconfig Command

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" in Chapter 3 for more information about disk identifiers.

- Determining how much space on each disk is available to the filer. Refer to "Understanding Usable Space on Each Disk" in Chapter 3 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 Dell custom MIB. For information about SNMP, see "Using SNMP" in Chapter 4.

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 Dell, provide the information displayed by `sysconfig -v`. This information is useful for diagnosing system problems.

Displaying Volume Information

Use the Vol status Command

The `vol status` command displays information about a volumes's configuration. The 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 volume states. This includes the volume name, whether it is on-line or off-line, and 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

Use the sysstat and uptime Commands

You use the `sysstat` and `uptime` commands to display filer statistics.

About 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:

- 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 "About 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 interface (1 MB per Ethernet adapter) you might need to add another network adapter card to the system.

About 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.

Example

An example of the display is

```
uptime

8:54am up 2 days 22:23, 3122520 NFS ops
```

Displaying Network Statistics

Use the netstat Command

You use the `netstat` command to display network statistics.

About the netstat Command

The `netstat` command displays network-related data in various output formats.

The `netstat -i` and `netstat -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" in chapter 18. For more information about the `netstat` command, see the *netstat*(1) man page.

Displaying Interface Statistics

Use the ifstat Command

The `ifstat` command prints per-interface statistics not reported by commands such as `netstat`. This includes some statistics maintained by the networking code, as well as statistics maintained by the driver and by the networking card.

The output of the `ifstat` command might contain many fields, because different types of interfaces, for example, Ethernet and Gigabit (GB) Ethernet have different statistics.

ifstat Syntax

The syntax for the `ifstat` command is

`ifstat [-z]  -a |` *interface*

The `-z` option "zeros" (or clears) the statistics. The `-a` option lists statistics for all the filer's interfaces. The interface option indicates the type of interface for which you want statistics.

Explanation of Interface Statistics

Ethernet

Table 17-1 describes the statistics in the RECEIVE section of the `ifstat` command output when you use the command on an Ethernet interface.

Table 17-1. ifstat Command on Ethernet Interface — RECEIVE

Statistic	Meaning
`Packets`	Number of packets received on the interface.
`Bytes`	Number of bytes received on the interface.
`Errors`	Number of errors during Ethernet frame reception, including all kinds of receive errors.
`No buffers`	Number of received packets dropped due to the unavailability of buffers.
`Length err`	Number of frames truncated due to the shortage of receive descriptors.
`Runt frames`	Number of runt frames.
`Long frames`	Number of frames received that exceeded the maximum Ethernet-specified size of 1,518 bytes.
`CRC error`	Number of CRC errors that occurred on the received frames.
`H/w overflow`	Number of frames discarded because of receive FIFO overflow.
`Process stop`	Number of times the receive process stopped.
`List overflow`	Number of frames dropped due to the unavailability of descriptors.
`Process reset`	Number of times the receive process was reset.
`Rst frame drops`	Number of frames discarded due to the resetting of the receive process.

Table 17-2 describes the statistics in the TRANSMIT section of the `ifstat` command output when you use the command on an Ethernet interface.

Table 17-2. ifstat Command on Ethernet Interface — TRANSMIT

Statistic	Meaning
`Packets`	Number of packets attempted to be transmitted.
`Bytes`	Number of bytes attempted to be transmitted.
`Errors`	Number of hardware errors encountered while attempting to transmit.
`Collisions`	Number of collisions that occurred while transmitting frames.
`Late collisions`	Number of collisions terminated due to a late collision.
`Excess coll`	Number of times transmission was terminated due to excessive collisions.
`Queue full`	Number of times the queue was full.
`List full`	Number of frames that were dropped due to the unavailability of descriptors.
`No carrier`	Number of times the carrier signal was not present during transmission.
`Underflow`	Number of times the transmitter terminated the message because data arrived late from memory.
`Defer`	Number of times transmission had to be deferred.
`Time out`	Number of times the transmit jabber timer expired.
`Stopped`	Number of times the receive process stopped.
`List underflow`	Number of frames that had to be dropped due to the unavailability of descriptors.
`Loss of carrier`	Number of times the carrier was lost.
`No buffers`	Number of times buffers were unavailable.
`Requeue`	Number of times the frame was requeued due to list underflow.
`Threshold up`	Number of times the transmit threshold was increased.
`Threshold dn`	Number of times the transmit threshold was decreased.
`Threshold lvl`	Current threshold level.

Table 17-3 describes the statistics in the DEVICE section of the `ifstat` command output when you use the command on an Ethernet interface.

Table 17-3. ifstat Command on Ethernet Interface — DEVICE

Statistic	Meaning
`Interrupts`	Number of times the Ethernet device interrupted the host.
`Resets`	Number of times the Ethernet device was reset.

Table 17-4 describes the statistics in the LINK INFO section of the `ifstat` command output when you use the command on an Ethernet interface.

Table 17-4. ifstat Command on Ethernet Interface — LINK INFO

Statistic	Meaning
`Auto`	Auto-Negotiation state.
`Mediatype`	Media type, such as twisted pair.
`Link Partner`	The Auto-Negotiation capability of the remote end. It is unknown if Auto-Negotiation is disabled.
`Link State`	Link status.

GB Ethernet

Table 17-5 describes the statistics in the RECEIVE section of the `ifstat` command output when you use the command on a GB Ethernet interface.

Table 17-5. ifstat Command on GB Ethernet Interface — RECEIVE

Statistic	Meaning
`Packets`	Number of packets received on the interface.
`Bytes`	Number of bytes received on the interface.
`Errors`	Number of errors during Ethernet frame reception, including all kinds of receive errors.
`Queue full`	Number of received packets dropped due to the unavailability of buffers.
`Unicast packets`	Number of unicast packets received.

Table 17-6 describes the statistics in the TRANSMIT section of the `ifstat` command output when you use the command on a GB Ethernet interface.

Table 17-6. ifstat Command on GB Ethernet Interface — TRANSMIT

Statistic	Meaning
`Packets`	Number of packets attempted to be transmitted.
`Bytes`	Number of bytes attempted to be transmitted.
`Errors`	Number of hardware errors encountered while attempting to transmit.
`Collisions`	Number of collisions that occurred while transmitting frames.
`Unicast packets`	Number of unicast packets transmitted.

Table 17-7 describes the statistics in the DEVICE section of the `ifstat` command output when you use the command on a GB Ethernet interface.

Table 17-7. ifstat Command on GB Ethernet Interface — DEVICE

Statistic	Meaning
`Received errors`	Number of errors encountered during reception by the interface.
`Transmit errors`	Number of errors encountered during transmission by the interface.
`Collisions`	Number of collisions encountered during transmission by the interface.

Improving Filer Performance

About This Section

This section 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 a `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 filer are attached to the same network where four clients reside, specify in */etc/fstab* on client1 and client2 that these clients mount from *filer-0:/home*. Specify in */etc/fstab* on client3 and client4 that these clients mount from *filer-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.

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 the `no_atime_update` option On. Consider turning this option On 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.

Improving Performance on Directory Lookups

Turning the `nfs.big_endianize_fileid` option On 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 the `big_endianize_fileid` *option On, 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.*

Improving Read-Ahead Performance

If the file access patterns of your clients are random (nonsequential), turning minimal read-ahead On might improve performance. By default, the filer uses aggressive read-ahead, which enhances sequential access, which is more commonly used by UNIX clients and applications. To specify minimal read-ahead, turn the `minra` option On. By default, the option is Off and the filer does very aggressive read-ahead.

CHAPTER 18
Troubleshooting

Getting Technical Assistance

Information to Note Before Calling for Support

If you encounter problems with the filer that you cannot solve, you might need to contact your service provider or Dell technical support for assistance. If you do, have the following information available:

- the system service tag and your Express Service code
- 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)

For additional information you may need, see "Getting Help" in your *Installation and Troubleshooting Guide*.

How to Contact Dell

For numbers that you can use to contact Dell technical support, see "Getting Help" in your *Installation and Troubleshooting Guide*.

Booting From System Boot Diskette

Boot From Diskette To Correct Some Types of Problems

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.

Procedure for Booting From Diskette

To display the boot menu from a diskette, perform the following steps:

1. Insert the diskette labeled System Boot Disk 1 into the filer's diskette drive.

NOTE: The Data ONTAP 5.3 system boot diskettes are specific to the model of filer that you are updating. You received diskettes appropriate to the model you are updating. If you have different models, be sure to check the diskette labels to make sure you're using the correct diskettes for the filer.

2. From the system console, enter the `reboot` command or, if the system is powered off, power it on.

   Result: The filer begins the boot process.

3. When the filer's LCD prompts you to, remove the diskette and insert the diskette labeled System Boot Disk 2 into the filer's diskette drive.

4. Press the Enter key on your console.

   Result: The filer boots and, if it can, displays the following boot menu:

```
1) Normal Boot

2) Boot without /etc/rc

3) Change Password

4) Initialize all disks

5) Maintenance mode boot

Selection (1-5)?
```

5. Choose one of the boot types shown below by entering the corresponding number.

   - Normal Boot (1) — Use Normal Boot to run the filer normally, but from a diskette.
   - Boot without */etc/rc* (2) — Use Boot without */etc/rc* to troubleshoot and repair configuration problems.

NOTE: Booting without /etc/rc causes the filer to use only default options settings, disregard all options settings you put in /etc/rc, and disable some services, such as `syslog`.

   - Change Password (3) — Use Change Password to reset your filer's administrative password.
   - Initialize all disks (4) — Use Initialize all disks to zero all disks attached to the filer.

NOTE: This action will result in a loss of all data on the disks.

- Maintenance mode boot (5) — 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 Shut Down Filer

Procedure for Restarting Filer After Unexpected Shutdown

Complete the following steps to restart your filer if it shuts down unexpectedly:

1. Write down the messages displayed on the console and the message on the LCD, if your filer has an LCD.
2. If the console is...

   Then...

   - Showing the `mon>` prompt — Enter **`b`** (for boot) to reboot the filer.
   - Showing the `ok` prompt — Enter **`boot`** to reboot the filer.
   - Unresponsive — Reset the filer by turning it off, leaving it off for 30 seconds, and turning it back on.

     Results: The filer boots and displays the system prompt on the console.
3. If the filer still does not boot, display the boot menu from diskette, as described in "Booting From System Boot Diskette," then choose Normal Boot.

NVRAM Problem

How the Filer Handles 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:

Inconsistency Due to Improperly Updated Volume

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.

Inconsistency Due to Log Updates for Off-line Volume

If the inconsistency is due to the log having updates for an off-line volume, the filer asks whether to discard them.

Inconsistency Due to Other Reasons

If there are many inconsistencies that cannot be repaired, the filer discards the inconsistent contents and creates a core dump file. 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

Types of Volume Problems Described

This section describes the following types of volume problems:

- Failed mounts and stale file handles
- Volume name problems

Failed Mounts and Stale File Handles

Changing Volume Names Can Cause Mount and File Handle Problems

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.

Procedure for Fixing the /etc/exports Problem

To fix the problem, edit the */etc/exports* file on the filer default volume to change the old volume name to the new volume name.

Results: The problem is resolved.

Volume Name Problems

Volume Naming Rules

A valid volume name has the following characteristics:

- 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 Volume Names

Examples of valid volume names are

- *_tech_pubs*
- *SW_Engineering*
- *Dept_32*

Error Messages About Volume Names

You might get an error message that contains one of the following phrases:

- invalid volume name
- unrecognized volume name
- illegal volume name

If you get one of these messages, take one of the following actions:

- Type correctly the name of an existing volume.
- Type a valid volume name.

Disk Problems

Types of Disk Problems Described

The next two sections describe how the filer reacts to a:

- Disk failure without a hot spare disk
- Disk failure with a hot spare disk

Disk Failure Without a Hot Spare Disk

About This Section

This section describes how the filer reacts to a disk failure when a hot spare disk is available.

Filer Runs in Degraded Mode

If a disk fails, the filer continues to run without losing any data but has a somewhat degraded performance.

CAUTION: Replace the disk as soon as possible, because a second disk failure could cause the filer to lose the entire file system.

Filer Logs Warning Messages in /etc/messages

The filer logs a warning message in the */etc/messages* file on the root volume every hour after a disk fails.

Filer Shuts Down Automatically After 24 Hours

To ensure that you notice the failure, the filer automatically shuts itself off in 24 hours, by default, or in a period you 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.

CAUTION: Check the */etc/messages* file on the root volume once a day for important messages. You can automate checking this file with a script on a remote host that periodically searches the file and then alerts you.

Filer Reconstructs Data After Disk Is Replaced

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

About This Section

This section describes how the filer reacts to a disk failure when a hot spare disk is not available.

Filer Replaces Disk With Spare and Reconstructs Data

If a disk fails, the filer

- replaces the failed disk with a hot spare disk
- reconstructs the missing data on the hot spare disk in the background, so that the interruption to file service is minimized
- logs the activity in the */etc/messages* file on the root volume

The filer does not shut down automatically.

CAUTION: 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" in Chapter 3.

If a second disk fails and there is no hot spare disk available, contact Dell technical support, as described in "Getting Technical Assistance."

Related Information

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 *sysconfig*(1) and *messages*(5) man pages.

Disk Errors

Types of Disk Errors Described

The filer displays error messages when the following disk problems occur:

- A disk does not exist.
- A disk is in use.
- Disks are missing.

Error Message: 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 disk is specified correctly.
- The specified disk is a spare.

Error Message: 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 disk is specified correctly.
- The specified disk is a spare.

Error Message: System Cannot Boot Because Disks Are Missing

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 disks because either not all the disks 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. Complete the following steps to resolve the problem.

1. Display the boot menu from diskette, as described in "Booting From System Boot Diskette," then choose Maintenance mode boot.
2. Use the `vol offline` command to take the volume off-line.
3. Reboot the filer; the volume specified in Step 2 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.

Inconsistent File System

Inconsistencies Seldom Occur

The file system rarely becomes inconsistent. However, an inconsistent file system can be a result of combined disk and NVRAM failure.

Contact Technical Support if an Inconsistency Occurs

If your file system becomes inconsistent, contact Dell technical support for assistance, as described in "Getting Technical Assistance."

Disk Operations in Maintenance Mode

Maintenance Mode Operations

Maintenance mode enables you to perform the following operations to troubleshoot disk problems:

- 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.

NOTE: For information about Maintenance mode, see "Booting From System Boot Diskette."

Displaying Detailed Disk Information

The `disk_list` command displays detailed device information for each disk on the system, such as 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.

CAUTION: 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, use this command to remove an old RAID label on a previously used disk that you are adding 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

The /etc/rc, /etc/exports, and /etc/hosts Files Can Contain Errors

Configuration problems usually occur in one of the three configuration files on the root volume:

- */etc/rc*
- */etc/exports*
- */etc/hosts*

This section describes common configuration problems.

What to Do When the Filer Is 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. Complete the following steps to fix this problem:

1. Log in to the filer from the system console.
2. At the filer prompt, enter the following commands, replacing the information shown in italics with values appropriate for your filer:

   ifconfig *if* mediatype *type IP_address* netmask *netmask*

   exportfs -i -o root=*adminhost_IP_addr* /

   nfs on

3. Mount the root file system from the administration host.
4. Edit the configuration files as described in Chapter 2, "Filer Administration Basics."

Filer Runs Setup When /etc/rc Is Damaged or Missing

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 System Boot Diskette" for information about booting from the system boot diskette.

How to Recover From Configuration Errors if NFS Is the Only Licensed Protocol

If you are running NFS only, complete the following steps to recover from configuration errors in the */etc/rc* file.

1. Display the boot menu from diskette as described in "Booting From System Boot Diskette."
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 do this procedure using CIFS.

3. At the filer prompt, enter the following two commands, replacing the variables shown in italics with values appropriate for your filer:

   ifconfig *if* mediatype *type IP_address* netmask *netmask*

   exportfs -i -o
   access=adminhost_IP_addr,root=adminhost_IP_addr nfs on

4. Mount the root file system from the administration host.
5. Edit the configuration files.
6. Remove the system boot diskette from the disk drive and reboot the filer to test the new configuration files.

NOTE: 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 if there is a power outage or system software failure.

How to Reset the Filer Password

Reset the Password if You Forget It

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.

Procedure for Resetting the Password

Complete the following steps to reset the filer password:

1. Reboot from diskette as described in "Booting From System Boot Diskette."
2. When the boot menu appears, enter **3** to choose Change Password.

3. When the filer prompts you for a new password, enter it at the prompt.

   Results: 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.

How to Initialize All Disks and Create a New File System

Initializing All Disks Erases All Data

You might need to initialize all disks and create a single new file system in the following circumstances:

- You decide to redeploy an existing filer and need to completely reconfigure it.
- Dell technical support advises you that the only way to recover from an error is to initialize all disks.

CAUTION: Initializing all disks causes all existing data to be lost.

Procedure for Initializing All Disks

Complete the following steps to initialize all disks:

1. If the console is...

   Then...

   - Displaying the filer prompt, for example, `filer>` — Place the system boot diskette into the diskette drive of the filer and enter **`reboot`**.
   - Not displaying the filer prompt — Reboot from diskette as described in "Booting From System Diskette."

2. When the boot menu appears, enter **4** to choose Initialize all disks.

   Results: The filer initializes all the disks and creates a single-volume file system.

Network Problems

Detect Network Problems Using ping at the Filer Console

You can detect network problems by going to the filer console and using the `ping` command.

What the ping Command Does

The `ping` command checks whether the filer can communicate with other hosts on the network and that other hosts can communicate with the filer.

How to Troubleshoot Network Problems

If the filer should be able to connect with a host but `ping` does not respond with a message indicating that the host is alive, complete the following steps to troubleshoot the problem:

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.

   NOTE: 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. Use the `routed status` command to determine 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 is 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 filer, create static mappings on your WINS servers for each of the filer's interfaces.

Contact Technical Support About Other Network Problem

If you encounter other problems, contact Dell technical support for assistance, as described in "Getting Technical Assistance."

NFS Problems

Client's Inability To Mount Directories Indicates 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 volumes or directories from the filer.

How to Troubleshoot NFS Problems

Complete the following steps to troubleshoot NFS problems:

1. Make sure that the filer is licensed for NFS by entering the `license` command at the filer prompt.

   If the following message appears

   ```
   nfs not licensed
   ```

   or if a message with protocols other than NFS appears but NFS is absent, you must get a license for NFS.

   NOTE: For information about how to get a license, contact Dell technical support, as described in "Getting Technical Assistance."

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.
5. Sometimes a client can see the filer but gets a Permission Denied message when requesting a mount. If this happens, follow these steps:

   - 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.
   - 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:

     ```
     options nfs.mount.rootonly off
     ```

     To make this change permanent, add the preceding command to your */etc/rc* file on the root volume.

Windows Access Problems

Kinds of Access Problems

This section describes preliminary troubleshooting steps, then describes how to troubleshoot the following problems:

- "Filer can't register with the Windows NT domain."
- "Incorrect password or unknown username."
- "Users can't map a drive."

Preliminary Troubleshooting Steps

Complete the following steps to begin to troubleshoot Windows problems:

1. Make sure that the filer is licensed for CIFS by entering the `license` command at the filer prompt.

   If the following message appears

   ```
   CIFS not licensed
   ```

   or if a message with protocols other than CIFS appears but CIFS is absent, you must get a license for CIFS.

NOTE: For information about how to get a license, contact Dell technical support, as described in "Getting Technical Assistance."

2. If you are authenticating through a Windows NT domain, make sure that the filer has an account in the domain.

NOTE: You can verify that the filer is registered with a domain controller by using the `cifs testdc` *command.*

3. Make sure that CIFS service has been properly configured with the `cifs setup` command.

Filer Can't Register With the Windows NT Domain

If you are using WINS, use the table "Using Wins" to troubleshoot the problem. If you are not using WINS, use the table "Not Using WINS."

Using WINS

Use Table 18-1 if you are using WINS.

Table 18-1. Using WINS

<table>
<tr><td colspan="5">Is the WINS server working?</td></tr>
<tr><td colspan="4">Yes</td><td>No</td></tr>
<tr><td colspan="3">Is the domain controller running?</td><td colspan="2" rowspan="7">Get the WINS server working.</td></tr>
<tr><td colspan="2">Yes</td><td>No</td></tr>
<tr><td colspan="2">Can you ping the domain controller?</td><td rowspan="6">Start the domain controller.</td></tr>
<tr><td>Yes</td><td>No</td></tr>
<tr><td>Can you ping the filer?</td><td rowspan="5">Check network connectivity</td></tr>
<tr><td>Yes</td><td>No</td></tr>
<tr><td>Contact Dell technical support, as described in “Getting Technical Assistance.”</td><td>Check network connectivity</td></tr>
</table>

Not Using WINS

Table 18-2 if you are not using WINS.

Table 18-2. Not Using WINS

<table>
<tr><td colspan="3">Is the PDC or BDC on the same subnet as the filer, and is the subnet connected to the first configured network interface card on the filer?</td></tr>
<tr><td colspan="2">Yes</td><td>No</td></tr>
<tr><td colspan="2">Is there a computer account created for the filer in the Windows NT domain?</td><td rowspan="3">Put the domain controller on the same subnet as the filer, and connect the subnet to the first configured network interface card on the filer.

Without a WINS server, the filer can talk only to the domain controller by broadcast. The filer broadcasts only from the first configured network interface. The domain controller must be on this subnet.</td></tr>
<tr><td>Yes</td><td>No</td></tr>
<tr><td>Contact Dell technical support, as described in “Getting Technical Assistance.”</td><td>Create a computer account for the filer in the domain.</td></tr>
</table>

Incorrect Password or Unknown Username

Use Table 18-3 to troubleshoot the problem.

Table 18-3. Incorrect Password or Unknown Username

<table>
<tr><td colspan="4">Are users in the same domain as the filer?</td></tr>
<tr><td colspan="2">Yes</td><td colspan="2">No</td></tr>
<tr><td colspan="2">Are you using a virtual LAN?</td><td colspan="2">Is there a trust relationship between the filer and the domain?</td></tr>
<tr><td>Yes</td><td>No</td><td>Yes</td><td>No</td></tr>
<tr><td>Have the users log in with DOMAIN-NAME\USERNAME</td><td>Enter the users into the /etc/passwd file</td><td>Have the users log in with DOMAIN-NAME\USERNAME</td><td>Establish a trust relationship between the user's account domain and the filer's domain</td></tr>
</table>

Users Cannot Map a Drive

If users get an Access Denied message, use the following decision table to troubleshoot the problem.

Table 18-4. Users Cannot Map a Drive

<table>
<tr><td colspan="3">Is the filer in the same domain as the PDC?</td></tr>
<tr><td colspan="2">Yes</td><td>No</td></tr>
<tr><td colspan="2">Does the user have access rights to the share?</td><td rowspan="3">Use Server Manager to add the filer to the domain or establish a trust relationship between the domains.</td></tr>
<tr><td>Yes</td><td>No</td></tr>
<tr><td>Contact Dell technical support, as described in "Getting Technical Assistance."</td><td>Give the user rights to the share.</td></tr>
</table>

UNIX cpio Problems

The cpio Version Should Support 32-bit Inode Definition Numbers

If you copy large amounts of data using the 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.

Why the Problem Occurs

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.

Ask UNIX Provider Whether cpio Version Supports 32-bit Inode Definition Numbers

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

The df Version Must Support Large File Systems

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.

Enable NFS Option to Avoid Displaying Useless Data

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."

DOS, Windows, and Macintosh Clients Might Have Display Problem

Some DOS, Windows, or Macintosh clients might have a display problem similar to UNIX systems; in these cases, enable the `nfs.v2.df_2gb_lim` option.

Filer df Command Always Shows Correct Disk Space

At all times, the `df` command entered on the filer correctly shows the amount of disk space used and remaining.

qtrees Affect Disk Space Displayed by df

If a directory in a qtree 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 qtree look fuller according to the client `df` command than it actually is.

Filer Quota Report Command Always Displays Correct Usage

The filer `quota report` command shows the correct usage within that qtree.

Serious Error Messages

Panic Messages Mean Serious Problems

If your filer has a serious problem, such as a problem with the hardware or a bug in the system software, it issues a system panic message similar to the following one:

```
PANIC: system hung (NS0)!
Volume: volname
Version: verno
```

Table 18-5 shows panic message components.

Table 18-5. Panic Message Components

Message Component	Description
`system hung (NS0)`	Indicates the panic class of the message and is significant. The actual text of the message varies with circumstances.
`volname`	Is the name of the volume.
`verno`	Is the version number.

What to Do After a Panic Message

Complete the following steps when your system issues a panic message:

1. Write down the panic message.
2. Call Dell technical support immediately, as described in "Getting Technical Assistance."
3. Provide the panic message to Dell technical support.

CHAPTER 19
Detailed Options Information

About options

About Setting Detailed Information

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

Option Values

The default value of an option is listed below the option name.

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

What the Autosupport Options Do

The autosupport options control whether and how the filer sends automatic status messages.

For more information about autosupport, see "Sending Automatic Email."

The autosupport.doit Option

Default

None

Description

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.

The autosupport.enable Option

Default

On

Description

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

The autosupport.from Option

Default

autosupport

Description

Specifies the sender of the automatic email message.

The autosupport. mailhost Option

Default

administration_host

Description

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

The autosupport.noteto Option

Default

None

Description

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

CIFS Options

What the CIFS Options Do

The CIFS options control CIFS features on the filer.

The cifs.access_logging_enable Option

Default

Off

Description

When On, enables the filer to process access logging, or auditing, information. The default is Off.

The cifs.access_logging.filename Option

Default

None

Description

Specifies the active event log file. The file must be in an existing directory in a network share.

The cifs.bypass_traverse_checking Option

Default

On

Description

When On, directories in the path to a file are not required to have the 'X' (traverse) permission.

NOTE: This option does not apply in UNIX qtrees or volumes.

The cifs.guest_account Option

Default

None

Description

When you leave `cifs.guest_account` blank, a CIFS user can log in to the filer without an account in the password database, provided that 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 and Generic Access" in Chapter 7.

The cifs.home_dir Option

Default

None

Description

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.

The cifs.idle_timeout Option

Default

18000

Description

Specifies the number of seconds that elapse before the filer disconnects an idle session. The value can range from 600 through 4,000,000 (effectively infinite).

The cifs.netbios_aliases Option

Default

None

Description

Specifies a list of alternative names for the filer. Use a comma-separated list of names.

The cifs.oplocks.enable Option

Default

On

Description

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.

The cifs.perm_check_use_gid Option

Default

On

Description

This option affects security checking for Windows clients of files with UNIX security where the requestor is not the file owner. In all cases, Windows client requests are checked against the share-level ACL. If the requestor is the owner, the User permissions determine the access.

If the requestor is not the owner and if perm_check_use_gid is On, files with UNIX security are checked using normal UNIX rules; that is, if the requestor is a member of the file's owning group, the Group permissions are used, otherwise the Other permissions are used.

The cifs.scopeid Option

Default

blank

Description

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.

This option is not recommended if you are using DNS for name resolution because NetBIOS Scope IDs and DNS are incompatible.

The cifs.search_domains Option

Default

None

Description:

Specifies a list of domains that trust each other to search for a mapped account. The argument for the option is a comma-separated list that is searched in order. If no list is supplied, all domains are searched. You use this option to limit searches if you used an asterisk for a domain name in the usermap.cfg file.

The cifs.show_snapshot Option

Default

FALSE

Description

Specifies whether to show the *~snapshot* directories in folders. To show the snapshot directories, set this option to TRUE.

The cifs.symlinks.cycleguard Option

Default

On

Description

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.

The cifs.symlinks.enable Option

Default

On

Description

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" in Chapter 5.

DNS Options

What the DNS Options Do

The DNS options control how the filer works with DNS.

For more information about DNS, see Chapter 4, "Network Administration."

The dns.domainname Option

Default

None

Description

Sets the DNS domain name to the specified domain name.

The dns.enable Option

Default

Off

Description

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

What the HTTP Options Do

The HTTP options enable and control HTTP services.

For more information about HTTP on the filer, see Chapter 8, "HTTP Administration."

The httpd.admin.enable Option

Default

On

Description

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

The httpd.enable Option

Default

Off

Description

Enables the HTTP server.

The httpd.log.max_file_size Option

Default

2147483647 (2 GB - 1 byte)

Description

Specifies the number of bytes */etc/log/httpd.log*, the HTTP log file, can grow to. The maximum value is 500 GB.

The httpd.rootdir Option

Default

None

Description

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

The httpd.timeout Option

Default

900 seconds (15 minutes)

Description

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

The httpd.timewait.enable Option

Default

On

Description

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 Options

What the NFS Option Does

The NFS option enables and controls NFS services.

For more information about NFS, see Chapter 6, "NFS Administration."

The nfs.mount_rootonly Option

Default

On

Description

Configures the filer to give NFS access only to root by requiring that mount requests come from privileged ports (ports 0 through 1023).

The nfs.per_client_stats.enable Option

Default

Off

Description

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

The nfs.tcp.enable Option

Default

Off

Description

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

The nfs.v2.df.2gb.lim Option

Default

Off

Description

Limits to 2 GB the response the filer gives to requests from NFS v2 clients 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.

The nfs.v3.enable Option

Default

On

Description

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.

The nfs.webnfs.enable Option

Default

Off

Description

Turns WebNFS On and Off.

The nfs.webnfs.rootdir Option

Default

None

Description

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.

The nfs.webnfs.rootdir.set Option

Default

FALSE

Description

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

NIS Options

What the NIS Options Do

The NIS options control how the filer works with NIS.

For more information about NIS, see Chapter 4, "Network Administration."

The nis.domainname Option

Default

None

Description

Sets the NIS domain to the specified domain name.

The nis.enable Option

Default

Off

Description

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

RAID Options

What the RAID Options Do

The RAID options control how the filer uses RAID.

For more information about RAID on the filer, see Chapter 3, "Disk and File System Management."

The raid.reconstruct_speed Option

Default

4

Description

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.

The raid.scrub.enable Option

Default

On

Description

Specifies whether the filer performs RAID scrubbing.

The raid.timeout Option

Default

24

Description

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

timed Options

What the timed Options Do

The `timed` options control whether and how the filer uses the `timed` daemon to synchronize time with a time server. For additional information about time synchronization and using the `timed` options, see "Filer System Time Synchronization" and "Synchronizing Filer System Time" in Chapter 2.

The timed.enable Option

Default

Off

Description

Determines whether a time daemon (`timed`) runs on the filer and synchronizes time with a time server.

The timed.log Option

Default

Off

Description

Determines whether to log to the console time changes initiated by the `timed` daemon.

The timed.max_skew Option

Default

30m

Description

Sets the maximum allowable discrepancy between filer time and server time. If there is a large discrepancy, it probably means that enough is wrong somewhere that the filer time should not be synchronized with the server time, no time synchronization takes place, and a message to that effect is sent to the console. The value is an integer followed by one of the following letters:

- s for seconds
- m for minutes
- h for hours

The default is 30 minutes.

The timed.proto Option

Default

ntp

Description

Selects whether to use the protocol used by the `rdate` command or SNTP. The value can be one of the following:

- rdate for the protocol used by the `rdate` command
- ntp for SNTP

The timed.sched Option

Default

hourly

Description

Schedules when to synchronize the time with a time server. The value can be one of the following:

- hourly to synchronize hourly
- multihourly to synchronize every six hours
- daily to synchronize every day at midnight
- a number followed by m to specify an interval of minutes or h to specify an interval of hours

To avoid overburdening the time server, the filer randomly selects the exact time of the synchronization within a 20-minute window of the specified schedule.

The timed.servers Option

Default

None

Description

Specifies up to five servers in order of contact priority. The value can be a list of the host names or IP addresses of up to five time servers, separated by commas. The filer goes down the list until it finds a server that responds, then uses the time from that server. It starts with the first server in the list each time. An example list of three time servers is as follows:

```
sundial, sundial.dell.com, 10.152.8.12
```

You can get a list of NTP (Network Time Protocol) time servers, which SNTP can use, from *http://www.eecis.udel.edu/~mills/ntp/servers.htm*.

volume Options

What the volume Options Do

The volume options control volume-level operations. You use these options only with the `vol options` command.

For more information about volume options, see "The vol options Command" in Chapter 2.

The Minra Option

Default

Off

Description

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

The no_atime_update Option

Default

Off

Description

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).

The nosnap Option

Default

Off

Description

Temporarily disables automatic snapshots.

The nosnapdir Option

Default

Off

Description

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.

The nvfail Option

Default

Off

Description

Sets the filer to check for NVRAM errors during boot up. Change the value to On when you want the filer to send error messages to notify you of NVRAM errors that can effect the validity of database files.

The raidsize Option

Default

None

Description

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

The root Option

Default

None

Description

Makes ***volume*** the root volume.

The snapmirrored Option

Default

None

Description

The filer automatically sets this option to On if the volume is a mirror for data replication. Otherwise, the option is set to Off. Change the value to Off if you want to convert a mirror to a regular volume. After you make the change, the volume is no longer read-only, and the filer stops making incremental changes to the volume for data replication.

NOTE: You cannot set this option to On. That is, you cannot use this option to convert a regular volume to a mirror. To use a volume as a mirror, follow the instructions in Chapter 16, "Data Replication Using SnapMirror," to start replicating data to the volume.

Miscellaneous Options

What the Miscellaneous Options Do

The miscellaneous options control additional aspects of filer operation.

The console.encoding Option

Default

nfs

Description

Table 19-1 specifies how non-ASCII character information is presented. The value can be one of the following.

Table 19-1. console.encoding Values

Value	Description
nfs	NFS character set. You can use both NFS extended (greater than 0x7F) and SGML characters for input.
sgml	SGML character format. You can use both NFS extended (greater than 0x7F) and SGML characters for input.
utf8	UTF-8 character sets. For input, any character greater than 0x7F is the beginning of a UTF-8 encoding.

The ip.match_any_ifaddr Option

Default

On

Description

If the option is On (the default), the filer accepts any packet that is addressed to it even if that packet came in on the wrong interface.

NOTE: If you are concerned about security, you should turn this Off.

The ip.path_mtu_discovery.enable Option

Default

On

Description

Enables or disables path MTU discovery, which is currently used only by TCP. When enabled, the filer can 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.

The rsh.enable Option

Default

On

Description

Enables the `rsh` server on the filer.

The snmp.enable Option

Default

On

Description

Enables the SNMP server on the filer.

The telnet.enable Option

Default

On

Description

Enables the Telnet server on the filer.

The telnet.hosts Option

Default

All hosts

Description

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 (-).

The vol.copy.throttle Option

Default

10

Description

Specifies the default speed of `vol copy` operations. The speed ranges from 10 (full-speed) to 1 (one-tenth of full-speed).

The wafl.convert_ucode Option

Default

Off

Description

Setting this option to On forces conversion of all directories to Unicode format when accessed from both NFS and CIFS. By default, conversion to Unicode format occurs as follows:

- Access from CIFS causes conversion of pre-4.0 and 4.0 format directories.
- Access from NFS causes conversion of 4.0 format directories.

The wafl.create_ucode Option

Default

Off

Description

Setting this option to On forces Unicode format directories to be created by default, both from NFS and CIFS. By default, all directories are created in pre-4.0 format and the first CIFS access converts a directory to Unicode format.

The wafl.default_nt_user Option

Default

None

Description

Specifies the Windows NT user account to use when a UNIX user accesses a file with Windows NT security (has an ACL), and that UNIX user would not otherwise be mapped. If this option is set blank, such accesses are denied.

The wafl.default_unix_user Option

Default

None

Description

Specifies the UNIX user account to use when a Windows NT user attempts to log in and that Windows NT user would not otherwise be mapped. If this option is set blank, such accesses are denied.

The wafl.maxdirsize Option

Default

10240

Description

Sets the maximum size, in kilobytes, of a directory file. A directory file with a size of 10,240 kilobytes can hold about 300,000 files or subdirectories.

The wafl.root_only_chown Option

Default

On

Description

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 non-root user changes the owner of a file, the set-user-id and set-group-id bits are cleared. If a non-root 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.

The wafl.wcc_minutes_valid Option

Default

20

Description

Specifies the number of minutes a WAFL credential cache entry is valid. The value can range from 1 through 20160.

APPENDIX A
Command Reference

This appendix provides the commands that you use to control a filer and are grouped in the following sections:

- User Commands
- File Formats
- Headers, Tasks, and Macros
- System Services and Daemons

User Commands

This section contains user commands.

NAME

arp - address resolution display and control

SYNOPSIS

arp *hostname*

arp -**a**

arp -**d** *hostname*

arp -**s** *hostname ether_address* [**temp**] [**pub**]

DESCRIPTION

The **arp** command displays and modifies the tables that the address resolution protocol uses to translate between Internet and Ethernet addresses.

With no flags, **arp** displays the current ARP entry for *host_name*. The host may be specified by name or by number, using Internet dot notation.

OPTIONS

-**a** Displays all of the current ARP entries.

-**d** Deletes an entry for the host called *hostname*.

-**s** Creates an ARP entry for the host called *hostname* with the Ethernet address *ether_address*. The Ethernet address is given as six hex bytes separated by colons. The entry will be permanent if the words following -**s** includes the keyword **temp**. Temporary entries that consist of a complete Internet address and a matching Ethernet address are flushed from the arp table if they haven't been referenced in the past 20 minutes. A permanent entry is not flushed.

If the words following -**s** include the keyword **pub**, the entry will be "published"; i.e., this system will act as an ARP server, responding to requests for *hostname* even though the host address is not its own.

SEE ALSO

ifconfig

NAME

cifs - summary of cifs commands

SYNOPSIS

Command Summary

This is a list of the subcommands of the **cifs** command.

cifs access	Modifies share-level Access Control List (ACL) entries.
cifs comment	Displays/modifies the CIFS server description.
cifs lookup	Translates user/group names into SIDs, and vice versa.
cifs restart	Restarts CIFS if it has been shut down with **cifs terminate**.
cifs setup	Configures CIFS service
cifs sessions	Displays current configuration and current connections.
cifs shares	Displays/modifies the CIFS exports.
cifs stat	Displays operational statistics.
cifs terminate	Shuts down CIFS, or logs off a single station.
cifs testdc	Tests the filer's connection to domain controllers.

SEE ALSO

cifs_access, cifs_comment, cifs_lookup, cifs_restart, cifs_setup, cifs_sessions, cifs_shares, cifs_stat, cifs_testdc, cifs_terminate

NAME

cifs access - modify share-level access control

SYNOPSIS

cifs access *share* [-**g**] *user rights*

cifs access -**delete** *share* [-**g**] *user*

DESCRIPTION

The **cifs access** command sets or modifies the share-level Access Control List ("ACL") of a share.

The *share* argument specifies the share whose ACL is to be modified. The *user* argument specifies the user or group of the ACL entry. *user* can be an NT user or group, if the filer is using NT domain authentication, or it can be a UNIX user or group, or it can be the special all-encompassing group **everyone**. The *rights* argument can be specified in either NT or UNIX style. NT-style rights are:

No Access

Read

Change

Full Control

UNIX-style rights are a combination of **r** for read, **w** for write, and **x** for execute.

If a share-level ACL entry for *user* already exists on the specified share, **cifs access** updates that ACL entry.

To display the current share-level ACL of a share, use Windows Server Manager or the **cifs shares** command.

OPTIONS

-g Specifies that *user* is the name of a UNIX group. Use this option when you have a UNIX group and a UNIX user or NT user or group with the same name.

-delete Deletes the ACL entry for *user* on *share*.

EXAMPLES

The following example grants NT Read access to the NT user **ENGINEERING\mary** on the share **releases**.

filer> **cifs access releases ENGINEERING\mary Read**

The following example grants UNIX read and execute access to the user **john** on the share **accounting**.

filer> **cifs access accounting john rx**

The following example grants full access to the UNIX group **wheel** on the share **sysadmins**.

filer> **cifs access sysadmins -g wheel Full Control**

The following example deletes the ACL entry for **ENGINEERING\mary** on the share **releases**.

filer> **cifs access -delete releases ENGINEERING\mary**

SEE ALSO

cifs_shares

NAME

cifs comment - display or change CIFS server description

SYNOPSIS

cifs comment [*newcomment*]

DESCRIPTION

The **cifs comment** command displays or changes the CIFS server description. CIFS clients see the CIFS server description when browsing servers on the network.

If no command-line arguments are given, **cifs comment** displays the current CIFS server description. If you enter a string for the *newcomment* parameter, the current CIFS server description is changed to *newcomment*. If *newcomment* contains spaces, enclose it in double quotation marks.

NAME

cifs lookup - translate name into SID or vice versa

SYNOPSIS

cifs lookup { *name* | *textualsid* }

DESCRIPTION

The **cifs lookup** command translates a Windows NT user or group name into its corresponding textual Windows NT SID (Security ID), or a textual NT SID into its corresponding Windows NT user or group name.

EXAMPLES

filer cifs lookup mday
SID = S-1-5-21-39724982-1647982808-1376457959-1221

filer cifs lookup NT-DOMAIN\mday
SID = S-1-5-21-39724982-1647982808-1376457959-1221

filer cifs lookup BUILTIN\Administrators
SID = S-1-5-32-544

filer cifs lookup S-1-5-32-544
name = BUILTIN\Administrators

filer cifs lookup nonexistantuser
lookup failed

NAME

cifs restart - restart CIFS service

SYNOPSIS

cifs restart

DESCRIPTION

cifs restart restarts CIFS service if it has been terminated by **cifs terminate**.

NAME

cifs sessions - information on current CIFS activity

SYNOPSIS

cifs sessions [*-s*] [*user*]

DESCRIPTION

The **cifs sessions** command displays information about CIFS users who are connected to the filer. If you omit the *user* argument, the command displays a summary of information about the filer and lists the users who are connected to the filer.

EXAMPLES

```
cifs sessions
Server Registers as 'HAWLEYR-TOKYO' in group 'NT-DOMAIN'
Filer is using ja for DOS users
WINS Server: 10.10.10.55
Selected domain controller \NT-DOM

PC (user)                                  #shares       #files
HAWLEY-PC       (hawleyr - root)             1             4
```

If you include the *user* argument, the command displays information about the specified user, along with the names and access level of files that *user* has opened. If you use ***** as the specified user, the command lists all users.

Executing the command for user **sam** might produce output as follows:

```
cifs sessions sam
users
  shares/files opened

HAWLEY-HOME1    (sam)
  ENG-USERS
     Read-denyW   - \SAM\SRC\FASWARE\PROD\COMMON\HTTPD\httpd_fast.c


HAWLEY-PC       (sam)
  ENG-USERS
```

The **-s** option displays security information for a specified connected user. If you do not specify a user or workstation name, the command displays security information for all users.

Executing the command **-s *** might produce the following:

```
cifs sessions -s *
users
  Security Information
```

```
WIN-95    (AGuest - nobody[guest])
        ***************
        UNIX uid = 1208
        user is a member of group nobody(65535)

        NT membership
                NT-DOMAIN\Domain Guests
                BUILTIN\Guests
        User is also a member of Everyone, Network Users
        ***************
```

NAME

cifs setup - configure CIFS service

SYNOPSIS

cifs setup

DESCRIPTION

The **cifs setup** command performs the initial configuration of the filer for CIFS. You must have installed the CIFS license before you enter this command. You must run the **cifs setup** command from the console or from a telnet connection; you can't enter the command through **rsh**.

FILES

/etc/cifsconfig.cfg	general configuration information
/etc/cifssec.cfg	NT domain machine account information
/etc/filersid.cfg	local machine SID **/etc/lclgroups.cfg** local NT group information
/etc/usermap.cfg	multiprotocol user map file

SEE ALSO

cifs_access, group, passwd

NAME

cifs shares - configure and display CIFS shares information

SYNOPSIS

cifs shares

cifs shares *sharename*

cifs shares -**add** *sharename path*

[-**comment** *description*]
[-**maxusers** *userlimit*]
[-**forcegroup** *groupname*]

cifs shares -**change** *sharename*

{ -**comment** *description* | -**nocomment** }
{ -**maxusers** *userlimit* | -**nomaxusers** }
{ -**forcegroup** *groupname* | -**noforcegroup** }

cifs shares -**delete** *sharename*

DESCRIPTION

cifs shares displays one or more shares, edits a specified share, creates a share, or deletes a share.

Listing shares

To list all shares and their access control lists, use the command **cifs shares** with no arguments. To list a single share and its access control list, use the command **cifs shares** *sharename* where *sharename* name of the share.

filer> **cifs shares**

```
Name          Mount Point           Description
----          -----------           -------
HOME          /vol/vol0/home        Default Share
                    everyone / Full Control
C$            /vol/vol0 Remote Administration
                    BUILTIN\Administrators / Full Control
ENGR          /vol/vol0/engr              Engineering
                    DOMAIN\Engineering / Full Control
NEWS          /vol/vol0/news              News
                    DOMAIN\Guests / No Access
                    everyone / Read
```

filer> **cifs shares news**

```
Name            Mount Point           Description
----            -----------  ----------
NEWS            /vol/vol0/news        News
                                      DOMAIN\Guests / No Access
                                      everyone / Read
```

Creating new shares

To create a new share, use the -**add** option:

cifs shares -**add** *sharename path*

[-**comment** *description*]
[-**maxusers** *userlimit*]
[-**forcegroup** *groupname*]

sharename name of the new share; clients use this name to access the share.

path full path name of the directory on the filer that corresponds to the root of the new share.

-**comment** *description*

description of the new share. CIFS clients see this description when browsing the filer's shares. If the description includes spaces, it must be enclosed in double quotation marks. If you do not specify a description, the description is blank.

-**maxusers** *userlimit*

maximum number of simultaneous connections to the new share. *userlimit* must be a positive integer. If you do not specify a number, the filer does not impose a limit on the number of connections to the share.

-**forcegroup** *groupname*

name of the group to which files to be created in the share belong. The *groupname* is the name of a group in the UNIX group database.

Deleting existing shares

To delete a share, use the -**delete** option:

cifs shares -delete "*sharename*"

sharename is the name of the share to be deleted. A share cannot be deleted if it is in use.

Changing the settings of existing shares
To change the settings of an existing share, use the **-change** option:

cifs shares -**change** *sharename*

{ -**comment** *description* | -**nocomment** }
{ -**maxusers** *userlimit* | -**nomaxusers** }
{ -**forcegroup** *groupname* | -**noforcegroup** }

The settings of a share can be changed at any time, even if the share is in use.

sharename is the name of the existing share that is to be changed.

-**comment** *description*

changes the description of the share. For more information about the share description setting, see the **Creating new shares** section, above.

-nocomment

changes the description of the share to an empty string.

-**maxusers** *userlimit*

changes the user limit on the share. For more information about the user limit setting, see the **Creating new shares** section, above.

-nomaxusers removes the user limit on the share.

-**forcegroup** *groupname*

changes the forcegroup setting. For more information about the forcegroup setting, see the **Creating new shares** section, above.

-**noforcegroup**

specifies that files to be created in the share do not belong to a particular UNIX group. That is, each file belongs to the same group as the owner of the file.

SEE ALSO

cifs_access

NAME

cifs stat - print CIFS operating statistics

SYNOPSIS

cifs stat [*interval*]

DESCRIPTION

The **cifs stat** command has two forms. If you specify the interval, the command continues displaying a summary of CIFS activity until interrupted. The information is for the preceding *interval* seconds. (The header line is repeated in the display every 10 lines.)

If you do not specify the interval, the command displays counts and percentages of all CIFS operations.

EXAMPLE

filer> **cifs stat 10**

GetAttr	Read	Write	Lock	Open/Cl	Direct	Other
175	142	3	70	115	642	50
0	0	0	0	0	18	0
0	8	0	0	3	8	0
0	10	0	0	0	0	0
0	6	0	0	1	0	0
0	0	0	0	0	0	0

NAME

cifs terminate - terminate CIFS service

SYNOPSIS

cifs terminate [-**t** *minutes*] [*workstation*]

DESCRIPTION

The **cifs terminate** command is used to terminate CIFS service. If the *workstation* operand is specified, then all CIFS sessions open by that workstation will be terminated. If no *workstation* operand is specified, then all CIFS sessions will be terminated and CIFS service will be shut down completely. To restart CIFS service after it has been shut down, use the **cifs restart** command (see **cifs_restart**).

If CIFS service is terminated for a workstation that has a file open, then that workstation will not be able to save any changes that it may have cached for that file, which could result in the loss of data. Therefore, it is *very* important to warn users before terminating CIFS service. The **-t** option, described below, can be used to warn users before terminating CIFS service.

If you run **cifs terminate** without the **-t** option and the affected workstations have open files, then you'll be prompted to enter the number of minutes that you'd like to delay before terminating. If you execute **cifs terminate** from **rsh** you will be required to supply the **-t** option since commands executed with **rsh** are unable to prompt for user input.

OPTIONS

-t *minutes* Specifies the number of minutes to delay before terminating CIFS service. During the delay the system will periodically send notices of the impending shutdown to the affected workstations. (Note: workstations running Windows95/98 or Windows for Workgroups won't see the notification unless they're running **WinPopup**.) If the specified number of minutes is zero, then CIFS service will be terminated immediately.

SEE ALSO

cifs_restart, halt, reboot, rsh

NAME

cifs testdc - test the filer's connection to Windows NT domain controllers

SYNOPSIS

cifs testdc

DESCRIPTION

The **cifs testdc** command tests the filer's ability to connect with Windows NT domain controllers. The output of the **cifs testdc** command is useful in the diagnosis of CIFS-related network problems.

EXAMPLE

purple> **cifs testdc**

```
Using Established configuration
Current Mode of NBT is H Mode

 NetBIOS scope ""
 Registered names...
      PURPLE        <0> WINS  Broadcast
      PURPLE        <3> WINS  Broadcast
      PURPLE        <20> WINS  Broadcast
      PURPLE-1       <0> WINS  Broadcast
      PURPLE-1      <3> WINS  Broadcast
      PURPLE-1      <20> WINS  Broadcast
      PURPLE-2       <0> WINS  Broadcast
      PURPLE-2       <3> WINS  Broadcast
      PURPLE-2       <20> WINS  Broadcast
      PURPLE-3       <0> WINS  Broadcast
      PURPLE-3       <3> WINS  Broadcast
      PURPLE-3       <20> WINS  Broadcast
      PURPLE-4       <0> WINS  Broadcast
      PURPLE-4       <3> WINS  Broadcast
      PURPLE-4       <20> WINS  Broadcast
      PURPLE-5       <0> WINS  Broadcast
      PURPLE-5       <3> WINS  Broadcast
      PURPLE-5       <20> WINS
      PURPLE-6       <0> WINS
      PURPLE-6       <3> WINS
      PURPLE-6       <20> WINS
      PURPLE-7       <0> WINS
      PURPLE-7       <3> WINS
      PURPLE-7       <20> WINS
      PURPLE-8       <0> WINS
      PURPLE-8       <3> WINS
      PURPLE-8       <20> WINS
      PURPLE-9       <0> WINS
      PURPLE-9       <3> WINS
      PURPLE-9       <20> WINS
```

```
NT-DOMAIN       <0> WINS
NT-DOMAIN       <3> WINS
NT-DOMAIN       <20> WINS

Testing Primary Domain Controller
found 2 addresses
trying 192.168.2.14...192.168.2.14 is alive
trying 192.168.2.85...192.168.2.85 is alive
found PDC NT-DOMAIN-BDC

Testing all Domain Controllers
found 4 addresses
trying 192.168.2.14...192.168.2.14 is alive
trying 192.168.2.85...192.168.2.85 is alive
trying 198.95.227.75...198.95.227.75 is alive
trying 192.168.2.14...192.168.2.14 is alive
found DC NT-DOMAIN-BDC
found DC FRENCH40
found DC NT-DOMAIN-BDC
found DC FRENCH40
```

NAME

date - display or set date and time

SYNOPSIS

date [**-u**] [[[[[*cc*]*yy*]*mm*]*dd*]*hhmm*[**.ss**]]

DESCRIPTION

date displays the current date and time when invoked without arguments.

When invoked with an argument, **date** sets the current date and time; the argument for setting the date and time is interpreted as follows:

cc	First 2 digits of the year (e.g., 19 for 1999).
yy	Next 2 digits of year (e.g., 99 for 1999).
mm	Numeric month. A number from 01 to 12.
dd	Day, a number from 01 to 31.
hh	Hour, a number from 00 to 23.
mm	Minutes, a number from 00 to 59.
ss	Seconds, a number from 00 to 59.

If the first 2 digits of the year are omitted, they default to 19; if all 4 digits of the year are omitted, they default to the current year. If the month or day are omitted, they default to the current month and day, respectively. If the seconds are omitted, they default to 0.

Time changes for Daylight Saving and Standard time, and for leap seconds and years, are handled automatically.

OPTIONS

-u Display or set the date in GMT (universal time) instead of local time.

EXAMPLES

To set the current time to 21:00:

```
date 2100
```

To set the current time to 21:00, and the current day to the 6th of the current month:

```
date 062100
```

To set the current time to 21:00, and the current day to December 6th of the current year:

```
date 12062100
```

To set the current time to 21:00, and the current day to December 6th, 1999:

date 9912062100

To set the current time to 21:00, and the current day to December 6th, 2002:

date 200212062100

SEE ALSO

rdate, timezone

NAME

df - display free disk space

SYNOPSIS

df [**-i**] [*pathname*]

DESCRIPTION

df displays statistics about the amount of free disk space in one or all volumes on the filer. All sizes are reported in 1024-byte blocks.

The *pathname* parameter is the path name to a volume. If it is specified, **df** reports only on the corresponding volume; otherwise, it reports on every on-line volume.

For each volume, **df** displays statistics about snapshots on a separate line from statistics about the active file system. The snapshot line reports the amount of space consumed by all the snapshots in the system. Blocks that are referenced by both the active file system and by one or more snapshots are counted only in the active file system line, not in the snapshot line.

If snapshots consume more space than has been reserved for them by the **snap reserve** command (see **snap**), then the excess space consumed by snapshots is reported as used by the active file system as well as by snapshots. In this case, it may appear that more blocks have been used in total than are actually present in the file system.

With the **-i** option, **df** displays statistics on the number of free inodes.

EXAMPLES

The following example shows file system disk space usage:

```
filer> df
Filesystem              kbytes   used    avail   capacity Mounted on
/vol/vol0              4339168  1777824 2561344 41%      /vol/vol0
/vol/vol0/.snapshot    1084788  956716   128072  88%      /vol/vol0/.snapshot
```

If snapshots consume more than 100% of the space reserved for them, then either the snapshot reserve should be increased (using **snap reserve**) or else some of the snapshots should be deleted (using **snap delete**). After deleting some snapshots, it may make sense to alter the volume's snapshot schedule (using **snap schedule**) to reduce the number of snapshots that are kept on line.

The following example shows file system inode usage for a specified volume:

```
filer df -i /vol/vol0
Filesystem        iused    ifree   %iused  Mounted on
/vol/vol0         164591   14313    92%    /vol/vol0
```

You can increase the number of inodes in a file system at any time using the **maxfiles** command (see **maxfiles**).

SEE ALSO

maxfiles, rc, snap

NAME

disk - RAID disk configuration control commands

SYNOPSIS

disk fail *disk_name*

disk remove *disk_name*

disk scrub start

disk scrub stop

disk swap

disk unswap

DESCRIPTION

The **disk fail** command forces a file system disk to fail; the **disk remove** command unloads a spare disk so that you can physically remove the disk from the filer. The **disk scrub** command causes the filer to scan disks for media errors. If a media error is found, the filer tries to fix it by reconstructing the data from parity and rewriting the data. Both commands report status messages when the operation is initiated and return completion status when an operation has completed.

The filer's "hot swap" capability allows removal or addition of disks to the system with minimal interruption to file system activity. Before you physically remove or add a SCSI disk, use the **disk swap** command to stall I/O activity. After you removed or added the disk, file system activity automatically continues. If you should type the **disk swap** command accidentally, or you choose not to swap a disk at this time, use **disk unswap** to cancel the swap operation and continue service.

If you want to remove or add a fibre channel disk, there is no need to enter the **disk swap** command.

Before you swap or remove a disk, it's a good idea to run **syconfig -r** to verify which disks are where.

USAGE

disk swap and disk unswap

applies to SCSI disks only and does not apply to PV filers.

disk fail *disk_name*

removes the specified file system disk from the RAID configuration, spinning the disk down when removal is complete. **disk fail** is used to remove a file system disk that may be logging excessive errors and requires replacement.

Note that when a file system disk has been removed in this manner, the RAID group to which the disk belongs will enter degraded mode (meaning a disk is missing from the RAID group). If a spare disk at least as large as the

disk being removed is available, the contents of the disk being removed will be reconstructed onto that spare disk.

The disk being removed is marked as "broken", so that if it remains in the disk shelf, it will not be used by the filer as a spare disk, and if it is moved to another filer, it will not be used by that filer as a spare disk.

disk remove *disk_name*

removes the specified spare disk from the RAID configuration, spinning the disk down when removal is complete. You can use **disk remove** to remove a spare disk so that it can be used by another filer (as a replacement for a failed disk or to expand file system space).

disk scrub start

starts a RAID scrubbing operation on all RAID groups. The **raid.scrub.enable** option is ignored; scrubbing will be started regardless of the setting of that option (the option is applicable only to scrubbing that gets started periodically by the system).

disk scrub stop

stops a RAID scrubbing operation.

SEE ALSO

sysconfig

NAME

disk_fw_update - update disk firmware

SYNOPSIS

disk_fw_update [*disk_name*]

DESCRIPTION

Use the **disk_fw_update** command to update out-of-date firmware on all disks or a specified disk on a filer. Each filer is shipped with a **/etc/disk_fw** directory that contains the latest firmware revisions. This command makes disks inaccessible for up to 2 minutes, so network sessions using the filer should be closed down before running it. This is particularly true for CIFS sessions, which will normally be terminated while this command executes.

Warning messages for disks being updated should be ignored while this command executes.

In the **/etc/disk_fw** directory, the firmware file name is in the form of *product_ID.revision.LOD*. For example, if the firmware file is for Seagate disks with product ID ST118202FC and the firmware revision is FD9E, the file name is **ST118202FC.FD9E.LOD**. The *revision* in the file name is the number against which the filer compares each disk's existing firmware revision. In this example, if the filer has disks with firmware revision F307, the file **/etc/disk_fw/ ST118202FC.FD9E.LOD**, assuming it exists, will be downloaded to all the disks when you execute this command.

To download the firmware to all disks, enter **disk_fw_update** without any arguments. To download the firmware to a particular disk, specify the disk name in the command, which is in the form of *adapter_number.disk_ID*. For example, if the disk ID is 1 and the disk is on adapter 8, enter the following command:

disk_fw_update 8.1

To determine disk firmware revisions, enter the **sysconfig -v** command. The following example is partial output from the **sysconfig -v** command. In this example, the firmware revision for the disk is FD9E.

```
slot 8: FC adapter: isp2100 (chip rev. 3)
    Firmware rev:  1.14.19
    Host Loop Id:  119
    FC Node Name:  2:000:00e08b:00a002
    Cacheline size:     8    FC Packet size:     34 000 000
    0: SEAGATE ST118202FC       FD90 Size=17GB (17783112 blocks)
```

NAME

download - install new version of Data ONTAP 5.3

SYNOPSIS

download

DESCRIPTION

download copies Data ONTAP 5.3 executable files from the **/etc/boot** directory to the filer's boot block on the disks from which the filer boots.

To install a new version of Data ONTAP 5.3, extract the files for the new release onto the filer from either a CIFS or an NFS client that has write access to the filer's root directory. For more information about how to install files for the new release, see the upgrade instructions that accompany each release.

After the filer reboots, you can verify the version of the newly installed software with the **version** command.

FILES

/etc/boot	directory of Data ONTAP 5.3 executables
/etc/boot/netapp-alpha	symbolic link to current version of Data ONTAP 5.3 for filers with Alpha processors
/etc/boot/fc-hard-alpha	boot FCode for filers with Alpha processors
/etc/boot/1-alpha	second stage boot code for filers with Alpha processors

SEE ALSO

version, boot

NAME

dump - file system backup

SYNOPSIS

dump [*options* [*arguments*]] *subtree*

DESCRIPTION

The **dump** command examines files in a subtree and writes to tape the files that need to be backed up. The Data ONTAP 5.3 **dump** command differs slightly from the standard UNIX **dump**, but the output format is compatible with SunOS 4.x/ Solaris 1.x and SunOS 5.x/Solaris 2.x **dump**.

Data ONTAP 5.3 **dump** can write to its standard output (most useful with **rsh** from a UNIX system), to a remote tape device on a host that supports the **rmt** remote tape protocol or to a local tape drive, connected directly to the system (**tape**).

The *subtree* argument specifies a subtree to be dumped. This is one way to allow **dump** to work with remote tape devices that are limited to 2 GB of data per tape file. The specified *subtree* may be in the active file system (e.g. **/home**) or in a snapshot (e.g. **/.snapshot/weekly.0/home**). If the subtree is in the active file system, **dump** creates a snapshot named **snapshot_for_dump.X** where X is a sequentially incrementing integer. This naming convention prevents conflicts between concurrently executing dumps. The dump is run on this snapshot so that its output will be consistent even if the filer is active. If **dump** does create a snapshot, it automatically deletes the snapshot when it completes.

Another way to allow **dump** to work with remote tape devices that are limited to 2 GB of data per tape file is to dump to multiple tape files or "volumes". The **B** option to **dump** specifies the maximum amount of data to be dumped to one volume; when that much data has been written to one volume, **dump** will start writing to another volume. A list of tape devices can be specified as arguments to the **f** option, and the volumes will be written to the devices in that list, in order. If there are no more devices in the list, **dump** will re-use the last device in the list, after prompting the user to indicate that they've put a new tape in that device.

The Data ONTAP 5.3 **dump** command handling of end of tape is slightly different than that of the standard UNIX dump. Instead of aborting the entire dump if EOT is reached, the Data ONTAP5.3 **dump** starts a new volume and continues the dump on the next tape. Specifying a large value for the **B** option will cause dump to utilize the entire tape.

You can enter the **dump** command on a trusted host through **rsh**. It is preferable to enter the **dump** command through **rsh** if the backup takes a significant amount of time. This is because if you enter the **dump** command on the console, the filer does not display the console prompt until the backup is finished. During the time when the filer is backing up data, you do not have console access to the filer.

Another advantage of running the **dump** command through **rsh** is that you can control backups from UNIX shell scripts or crontab entries.

OPTIONS

0-**9** Dump levels. A level 0, full backup, guarantees the entire file system is copied. A level number above 0, incremental backup, tells dump to copy all files new or modified since the last dump of a lower level. The default level is 0.

f *files* Write the backup to the specified *files*. *files* may be:

a list of the names of local tape devices, in the form specified in **tape**;

a list of the names of tape devices on a remote host, in the form *host:devices*;

the standard output of the dump command, specified as -.

The list may have a single device or a comma-separated list of devices; note that the list must either contain only local devices or only devices on a remote host and, in the latter case, must refer to devices on one particular remote host, e.g.

tapemachine:/dev/rst0,/dev/rst1

Each file in the list will be used for one dump volume in the order listed; if the dump requires more volumes than the number of names given, the last file name will used for all remaining volumes after prompting for media changes.

Use **sysconfig** -**t** for a list of local tape devices. See below for an example of a dump to local tape.

For a dump to a tape device on a remote host, *host* must support the standard UNIX **rmt** remote tape protocol.

By default, **dump** writes to standard output.

B *blocks* Set the size of the dump file to the specified number of 1024-byte blocks, If this amount is exceeded, **dump** will close the current file and open the next file in the list specified by the **f** option. If there are no more files in that list, **dump** will re-open the last file in the list, and prompt for a new tape to be loaded.

It is recommended to be a bit conservative on this option.

This is one way to allow **dump** to work with remote tape devices that are limited to 2 GB of data per tape file.

u Update the file **/etc/dumpdates** after a successful dump. The format of **/etc/dumpdates** is readable by people. It consists of one free format record per line: subtree, increment level and **ctime** format dump date. There may be only one entry per subtree at each level. The dump date is defined as the creation date of the snapshot being dumped. The file

/etc/dumpdates may be edited to change any of the fields, if necessary. See **dumpdates** for details.

b *factor* Set the tape blocking factor in k-bytes. The default is 63 KB. If the density is set to greater than 6250 BPI, then the default blocking factor is 32 KB. **NOTE:** Some systems support blocking factors greater than 63 KB by breaking requests into 63-KB chunks or smaller using variable sized records; other systems do not support blocking factors greater than 63 KB at all. When using large blocking factors, always check the system(s) where the potential **restore** might occur to ensure that blocking factor specified in **dump** is supported. Local tape devices restrict the blocking factor to less than, or equal to, 63 KB.

l Specifies that this is a multi-subtree dump. The directory that is the common root of all the subtrees to be dumped must be specified as the last argument. The subtrees are specified by path names relative to this common root. The list of subtrees is provided from standard in, one item on each line, with a blank line to terminate the list. (If you use this option, you must also use option **n**.)

n Specifies the dumpname for a multi-subtree dump. Mandatory for multi-subtree dumps.

Q Backs up all files and directories in qtree 0 of the specified volume. Qtree 0 is a qtree that is not created by you with the **qtree** command. In each volume, the files and directories that do not belong to a qtree you create are considered to be in qtree 0. Follow the **Q** option with the path name of a volume (for example, **/vol/vol1**).

X Specifies an exclude list, which is a comma-separated list of strings. If the name of a file matches one of the strings, it is excluded from the backup. The following list describes the rules for specifying the exclude list:

The name of the file must match the string exactly.

An asterisk is considered a wildcard character.

The wildcard character must be the first or last character of the string. Each string can contain up to two wildcard characters.

If you want to exclude files whose names contain a comma, precede the comma in the string with a backslash.

You can specify up to 32 strings in the exclude list.

EXAMPLES

To make a level 0 dump of the entire file system to a remote tape device with each tape file in the dump being less than 2 GB in size, use:

```
filer> dump 0ufbB adminhost:/dev/rst0 63 2097151 /
```

To make a level 0 dump of **/home** on a 2 GB tape to a remote tape device, use:

filer> **dump 0ufbB adminhost:/dev/rst0 63 2097151 /home**

To make a level 0 dump of **/home** on a 2 GB tape to a local tape drive (no rewind device, unit zero, highest density) use:

filer> **dump 0ufbB nrst0a 63 2097151 /home**

To make a level 0 dump of the entire file system to a local tape drive (no rewind device, unit zero, highest density), with each tape file in the dump being less than 2 GB in size, without operator intervention, using a tape stacker, with four tape files written per tape, assuming that the dump requires no more than 10GB, use:

filer> **dump 0ufbB nrst0a,nrst0a,nrst0a,urst0a,rst0a 63 2097151 /**

This will:

write the first three files to the norewind device, so that they, and the next dump done after them, will appear consecutively on the tape;

write the next file to the unload/reload device. This will cause the stacker to rewind and unload the tape after the file has been written and then load the next tape.

write the last file to the rewind device, so that the tape will be rewound after the dump is complete.

To back up all files and directories in a volume named **engineering** that are not in a qtree you created, use:

filer> **dump 0ufQ rst0a /vol/engineering**

To run the **dump** command through **rsh**, enter the following command on a trusted host:

adminhost# rsh filer dump 0ufbB adminhost:/dev/rst0 63 2097151 /home

FILES

/etc/dumpdates

dump date record

SEE ALSO

quota, rshd, restore, snap, sysconfig, tape, dumpdates

NOTES

Restore

As stated previously, filer **dump** output format is compatible with SunOS 4.x/Solaris 1.x and SunOS 5.x/Solaris 2.x **dump**. The filer supports a local **restore** command (see **restore**), so the restoration process can be

performed on the filer. It can also be performed via a **restore** done on an NFS client machine; if such a restore is being done, the client system should be checked to ensure it supports SunOS-compatible **dump**/**restore** format.

Client Dump and Restore Capability

If a client is to be used for performing filer dump and/or restore, it is important to check what the maximum dump and restore capabilities of your client system are before setting up a dump schedule. There are some client systems which do not support dump and restore of greater than 2 GB while others may support very large dumps and restores. It is especially important to check the **restore** capability of your system when using the filer local tape dump since the filer supports dumps that are greater than 2 GB.

Tape Capacity and Dump Scheduling

Along with the potential 2-GB restriction of **dump** or **restore** on a client system, it is important to consider your tape capacity when planning a dump schedule. For the filer local tape option, the Exabyte 8505 supports an approximate maximum capacity of 10GB per tape using compression. If a client system is used as the target for your dump, the capacity of that tape drive should be checked for dump planning.

If your filer file system exceeds the capacity of the local tape drive or the client system dump/restore, or you choose to dump multiple file system trees to make the restore process with multiple tape drives parallel, you must segment your dump to meet these restrictions.

One way to plan a dump schedule with a UNIX client system is to go to the root mount point of your filer and use the **du** command to obtain sizes of underlying subtrees on your filer file system. Depending on the restrictions of your client's dump and restore capability or recording capacity of the tape device being used, you should specify a **dump** schedule that fits these restrictions. If you choose to segment your dump, the **norewind** device (see **tape**) can be used to dump multiple tape files to one physical tape (again, choose a dump size which meets the criteria of your client restore and capacity of your tape drive).

The following example shows the **du** output from a filer file system on a client that supports dump and restore that are greater than 2 GB:

```
client% du -s *
4108    etc
21608   finance
5510100 home
3018520 marketing
6247100 news
3018328 users
```

You can use a tape device with approximately 10 GB on each tape to back up this filer. The dump schedule for this system can use the **norewind** tape device to dump the *marketing* and *news* subtrees to one tape volume, then

load another tape and use the **norewind** tape device to dump *etc*, *finance*, *home* and *users* subtrees to that tape volume.

CIFS Data

The Data ONTAP 5.3 **dump** command dumps the CIFS attributes and 8.3 name data for each file that is backed up. This data will *not* be backed up by a dump run on an NFS client machine. This data will *not* be restored by a restore run on an NFS client machine. This data will only be restored if a local restore is done of a backup created by the Data ONTAP 5.3 **dump** command.

NAME

exportfs - export and unexport files or directories

SYNOPSIS

exportfs [-**aiuv**] [-**o** *options*] [*pathname*]

DESCRIPTION

If no *pathname* is specified, **exportfs** lists all currently exported directories and files. If *pathname* is specified, **exportfs** makes the specified file or directory available or unavailable for mounting by NFS clients.

OPTIONS

-**a** Takes the list of path names to be exported or unexported from the **/etc/exports** file. If you specify *pathname* in the command when using the **-a** option, the command ignores *pathname*.

-**i** Ignores the options in the **/etc/exports** file. Without the **-i** option, the **exportfs** command uses the options associated with the *pathname* specified in **/etc/exports**.

-**u** Unexports the specified path name. If you also include the **-a** option, the command unexports the path names in the **/etc/exports** file and ignores *pathname*.

-**v** Prints each path name as it is exported or unexported.

-o *option*
Specifies a list of comma-separated options that describe how a file or directory is exported. You can specify the option in one of the following formats:

access=*hostname*[:hostname]...
Give mount access to each host listed. Alternatively, you can specify a netgroup instead of a host in the list. The netgroup must be defined in the **/etc/netgroup** file. Whether the hosts can mount *pathname* with root access, read-and-write access, or read-only access depends on how you use the **root**, **rw**, and **ro** options, as described below.

anon=*uid*
If a request comes from user ID of 0 (root user ID on the client), use *uid* as the effective user ID unless the client host is included in the **root** option. The default value of *uid* is 65534. To disable root access, set *uid* to 65535. To grant root access to all clients, set *uid* to 0.

ro Export the *pathname* read-only. If you do not specify this option, the *pathname* is exported read-write.

rw=*hostname*[:hostname]...
Export the *pathname* read-only to all hosts not specified in the list and read-write to the hosts in the list. Netgroup names are not allowed in the list.

root=*hostname*[:hostname]...
Give root access only to the specified hosts. By default, no hosts are granted root access. Netgroup names are not allowed in the list.

When you export a file or directory using the **ro**, **rw**, or **root** option, you can specify that the file or directory be exported to a subnet instead of individual hosts. You cannot export to a subnet when using the **access** option.

Instead of specifying a host name or netgroup name in the **exportfs** command, specify the subnet in one of the following formats:

dotted_IP/*num_bits*
The *dotted_IP* field is either an IP address or a subnet number. The *num_bits* field specifies the size of the subnet by the number of leading bits of the netmask.

"[**network**] *subnet* [**netmask**] *netmask*"
The *subnet* field is the subnet number. The *netmask* field is the netmask.

In UNIX, it is illegal to export a directory that has an exported ancestor in the same file system. Data ONTAP 5.3 does not have this restriction. For example, you can export both the **/** directory and the **/home** directory. In determining permissions, the filer uses the longest matching prefix.

EXAMPLES

In the following example, all network clients can mount the **/home** directory but only the **adminhost** can mount the **/** directory:

```
exportfs -o access=adminhost,root=adminhost /home
exportfs /
```

The following examples show different forms of the **exportfs** command that export the **/home** directory to the 123.45.67.0 subnet with the 255.255.255.0 netmask:

```
exportfs -o rw=123.45.67.0/24 /home
exportfs -o rw=123.45.67/24 /home
exportfs -o rw="network 123.45.67.0 netmask 255.255.255.0"
exportfs -o rw="123.45.67.0 255.255.255.0"
```

FILES

/etc/exports	directories and files exported to NFS clients
/etc/hosts	host name data base
/etc/netgroup	network groups data base

SEE ALSO

exports, hosts, netgroup

NOTES

Data ONTAP 5.3 supports a maximum of 255 host names in each **rw** and **root** option. There is no limit on the number of host names in the list following the **access** option, but the maximum size of the **/etc/exports** file is about 64 KB.

NAME

fctest - test Fibre Channel environment

SYNOPSIS

fctest [*adapter*]

DESCRIPTION

Use the **fctest** command to test Fibre Channel adapters and disks on an appliance. This command provides a report of the integrity of your Fibre Channel environment. It is only available in maintenance mode, and takes about 5 minutes to complete.

If the *adapter* argument is missing, all Fibre Channel adapters and disks in the system are tested, otherwise only the specified adapter, and disks attached to it, are tested.

When finished, **fctest** prints out a report of the following values for each Fibre Channel adapter tested:

1. Number of times loss of synchronization was detected in that adapter's Fibre Channel loop.
2. Number of CRC errors found in Fibre Channel packets.
3. The total number of inbound and outbound frames seen by the adapter.
4. A "confidence factor" on a scale from 0 to 1 that indicates the health of your Fibre Channel system as computed by the test. A value of 1 indicates that no significant errors were found. Any value less than 1 indicates there are problems in the Fibre Channel loop that are likely to interfere with the normal operation of your appliance. There is a troubleshooting checklist for Fibre Channel problems in your *System Administrator and Command Reference Guide* that may help identify and correct the problem.

If the confidence factor is reported as less than 1, go through the troubleshooting checklist for Fibre Channel loop problems in the System Administrator and Command Reference Guide and re-run the **fctest** command after making any suggested modifications to your Fibre Channel setup.

The actual arithmetic that is used to compute the confidence factor is as follows:

The number of Fibre Channel frame errors is obtained by adding the number of CRC and Synchronization errors, with each sync error weighted ten times as much as a CRC error. The number of errors is then divided by the total number of Fibre Channel frames (inbound + outbound). The quotient is subtracted from 1 to get the final answer.

C = 1 - (Crc + 10Sync)/F,

where C = the confidence factor, representing the integrity of your FC system.

Crc = number of CRC errors observed in the Fibre Channel frames.

Sync = number of Fibre Channel synchronization errors.

F = the total number of inbound and outbound frames seen by the adapter.

NAME

filestats - collect file usage statistics

SYNOPSIS

filestats [ages *ages*] **[timetype** {*a,m,c,cr*}**] [sizes** *sizes*] **[snapshot** *snapshot_name*] **[style** *style*] **[volume** *volume_name*]

DESCRIPTION

The **filestats** utility provides a summary of file usage within a volume. It must be used on a snapshot, and the only required argument is the snapshot name. The volume name defaults to "vol0" if not specified. If the volume you are examining is named otherwise, specify the name explicitly.

OPTIONS

The following options are supported.

ages *ages*

Specifies the breakdown of ages, as a set of comma-separated time values. The values are in seconds, but as a convenience you can add an H or D suffix to a number to get hours and days. For example, "900,4H,7D" would produce a breakdown with 4 categories - files accessed in the last 15 minutes, files accessed in the last four hours, files accessed in the last week, and all other files.

expr *expression*

This lets you specify a boolean expression that will be evaluated for each inode encountered, and if the expression is true, then the inode will be selected and included in the various breakdowns of file usage. The expression can contain "variables", which are merely the name of an inode attribute enclosed in curly braces. For example, {size} is evaluated as the size of the current inode. The valid inode attributes that you can use in expressions are:

tid	The tree id (for qtrees).
type	The file type (numeric, currently).
perm	Permissions.
flags	Additional flags.
nlink	Count of hard links.
uid	User id (numeric) of file owner.
gid	Group id (numeric) of file owner.
size	Size in bytes.
blkcnt	Size in blocks.
gen	Generation number.

atime	Time of last read or write (in seconds).
mtime	Time of last write (in seconds).
ctime	Time of last size/status change (in seconds).
crtime	Time file was created (in seconds).
atimeage	Age of last read or write (Now - atime).
mtimeage	Age of last write (Now - mtime).
ctimeage	Age of last size/status change (Now ctime).
crtimeage	Age of file creation (Now - crtime).

timetype *timetype*

This lets you specify the type of time that will be used in the "age" comparison. Valid values for *time_type* are

a	Access time
m	Modification time
c	Change time (last size/status change)
cr	Creation time

sizes *sizes*

Specifies the breakdown of sizes, as a comma-separated set of size values. The values are in bytes, but as a convenience you can add a K, M, or G suffix to a number to get kilobytes, megabytes, and gigabytes. For example, "500K,2M,1G" would produce a breakdown with 4 categories - files less than 500K, files less than 2 megabytes, files less than 1 gigabyte, and all other files.

To produce a breakdown that includes all unique file sizes, specify "*" for the sizes value.

style *style*

Controls the style of output - the possible value for *count* are "readable" (the default), "table" (colonseparated values suitable for processing by programs), and "html".

EXAMPLES

1. Produce default file usage breakdowns for snapshot *hourly.1* of volume *vol0*.

   filestats volume vol0 snapshot hourly.1

2. Produce file usage breakdowns by monthly age values:

   filestats volume vol0 snapshot hourly.1 ages "30D,60D,90D,120D,150D,180D"

3. Produce file usage breakdowns for inodes whose size is less than 100000 bytes and whose access time is less than a day old:

   filestats volume vol0 snapshot hourly.1 expr "{size}<100000&&{atimeage}<86400)"

4. Produce a breakdown of the total number of files and their total size. You can control the set of ages and sizes that get used for this breakdown, with the "ages" and "sizes" arguments. The output also contains a breakdown of file usage by user-id and group-id.

   filestats snapshot hourly.1 volume vol0

NOTES

Currently, the expression-evaluating code does not do any optimizations, so although you can use arithmetic expressions, it is most efficient if you do not. Of course, it's most efficient if you don't use any expression at all.

NAME

halt - stop the filer

SYNOPSIS

halt [-**d**] [-**t** *mins*]

DESCRIPTION

halt flushes all cached data to disk and drops into the monitor.

NFS clients can maintain use of a file over a **halt** or **reboot** (although experiencing a failure to respond during that time), but CIFS clients cannot do so safely. Therefore, if the filer is running CIFS, the **halt** command invokes **cifs terminate**, which requires the -**t** option. If the filer has CIFS clients and you invoke **halt** without -**t**, it displays the number of CIFS users and the number of open CIFS files. Then it prompts you for the number of minutes to delay. **cifs terminate** automatically notifies all CIFS clients that a CIFS shut-down is scheduled in *mins* minutes, and asks them to close their open files. CIFS files that are still open at the time the filer halts will lose writes that had been cached but not written.

halt logs a message in **/etc/messages** to indicate that the filer was halted on purpose.

OPTION

-**d**	Dumps system core before halting.
-**t** *mins*	Halts after the indicated number of minutes, or after all CIFS files that were open have been closed, whichever is sooner.

SEE ALSO

cifs_terminate, reboot, savecore, messages

NAME

help - print summary of commands and help strings

SYNOPSIS

help [*command* ...]

? [*command* ...]

DESCRIPTION

help prints a summary for each command in its argument list. With no arguments, **help** prints a list of all available Data ONTAP 5.3 commands.

Full UNIX-style man pages for all filer commands and files are available in the **/etc/man** directory.

FILES

/etc/man directory of UNIX-style manual pages

NAME

hostname - set or display Dell filer name

SYNOPSIS

hostname [*name*]

DESCRIPTION

hostname prints the name of the current host. The hostname can be set by supplying an argument. This is usually done in the initialization script, **/etc/rc**, which is run at boot time. *name* must exist in the **/etc/hosts** data base.

FILES

/etc/hosts	host name data base
/etc/rc	system initialization command script

SEE ALSO

hosts, rc

NAME

httpstat - display HTTP statistics

SYNOPSIS

httpstat [-**tz**] [*interval*]

DESCRIPTION

httpstat displays statistical information about HTTP (HyperText Transfer Protocol) for the filer. It can also be used to reinitialize this information. If no arguments are given, **httpstat** displays statistical information since last reboot, or last zeroed with the -**z** option. If the -**t** option is specified, statistical information since the last reboot is given.

The output consists of the number of GET requests successfully processed (**gets**), rejected requests (**badcalls**), currently open HTTP connections (**open conn.**), and the maximum number of simultaneous connections (**peak conn.**).

If the *interval* argument is specified, **httpstat** will continuously display the summary information for all the statistics. The first line of data displayed contains cumulative statistics. Each subsequent line shows incremental statistics for the *interval* (in seconds) since the last display.

SEE ALSO

netstat, options, sysstat

NAME

ifconfig - configure network interface parameters

SYNOPSIS

ifconfig *interface* [[**alias** | -**alias**] *address*]

[**netmask** *mask*] [**broadcast** *address*]
[**mediatype** *type*] [**mtusize** *size*] [**up** | **down**]
[**trusted** | **untrusted**] [**wins** | -**wins**]

ifconfig -**a**

DESCRIPTION

ifconfig assigns an address to a network interface and configures network interface parameters. **ifconfig** must be used at boot time to define the network address of each network interface present on a machine; it may also be used at a later time to redefine a network interface's address or other operating parameters. When used without optional parameters, **ifconfig** displays the current configuration for a network interface.

The *interface* parameter is the name of the network interface. The name is of the form **en** for Ethernet interfaces, possibly followed by a letter, where *n* is **0** for onboard network interfaces and the expansion slot number for network interfaces plugged into expansion slots. If a card in an expansion slot has more than one network interface, the network interface name will be followed by a letter, indicating which of the network interfaces on that card it is. The network interface name **vh** is used to specify IP virtual host addresses associated with the filer. Only alias addresses (using the *alias* option) may be assigned to the **vh** interface. The network interface name -**a** is special and it does not take any optional parameters. It displays the current configuration for all the network interfaces present.

The *address* is either a host name present in the host name data base **/etc/hosts** or an Internet address expressed in the Internet standard dot notation.

OPTIONS

broadcast *address*
Specifies the address to use to represent broadcasts to the network. The default broadcast address is the address with a host part of all 1's.

down
Marks a network interface "down". When a network interface is marked "down" the system will not attempt to transmit messages through that network interface. If possible, the network interface will be reset to disable reception as well. This action does not automatically disable routes using the network interface.

mediatype *type*
Specifies the Ethernet media type used. Depending on the physical specifications of the Ethernet card the acceptable types are "thick" (10Base5 AUI), "thin" (10Base2 BNC), "tp" (10Base-T RJ-45 twisted-pair), or "tpfd" (Full

duplex 10Base-T RJ-45 twistedpair), or "100tx" (100Base-T RJ-45 twisted-pair), or "100tx-fd" (Full duplex 100Base-T RJ-45 twisted-pair), or "auto" (Auto RJ-45 twisted-pair). The default media type is set to "tp" or to "auto" where applicable.

On a 10/100 Mbps auto-negotiable interface, the system will auto-negotiate a 10 Mbps half or full duplex or 100 Mbps half or full duplex link and set the network interface accordingly when it is configured up. If the other end does not support auto-negotiation and full duplex operation is desired, it must be explicitly set using the **mediatype** command.

On a 10/100 Mbps interface, the system will auto-detect a 10 Mbps or 100 Mbps link and set the link speed accordingly when the network interface is configured up. The hardware is not currently capable of autodetecting full duplex interfaces, so if full duplex operation is desired, it must be explicitly set using the **mediatype** command. Only the 10/100 Mbps interfaces are capable of full duplex operation.

mtusize *size*
Specifies the MTU (maximum transmission unit) to use for the network interface.

netmask *mask*
The mask includes the network part of the local address and the subnet part, which is taken from the host field of the address. The mask can be specified as a single hexadecimal number with a leading 0x, with a dot-notation Internet address, or with a pseudo-network name listed in the network table **/etc/networks**. The mask contains 1's for the bit positions in the 32-bit address that are to be used for the network and subnet parts, and 0's for the host part. The mask should contain at least the standard network portion, and the subnet field should be contiguous with the network portion. A default *netmask* is chosen according to the class of the IP address.

up
Marks a network interface "up". This may be used to enable a network interface after an "ifconfig down." It happens automatically when setting the first address on a network interface. If the network interface was reset when previously marked down, the hardware will be re-initialized.

alias
Establishes an additional network address for this network interface. This is sometimes useful when changing network numbers and one wishes to accept packets addressed to the old network interface. It is required when creating IP virtual host addresses.

-alias
Remove a network address for this network interface.

trusted
Specifies that the network to which the network interface is attached is trusted relative to firewall-style security (default).

untrusted

Specifies that the network to which the network interface is attached is not trusted relative to firewall-style security.

wins

Specifies that the network interface is to be registered with Windows Internet Name Services (default). Such registration is only performed when CIFS is running and at least one WINS server has been configured.

-wins

Specifies that the network interface is not to be registered with Windows Internet Name Services.

SEE ALSO

hosts, networks

NAME

ifstat - display device-level statistics for network interfaces

SYNOPSIS

ifstat [**-z**] **-a** | *interface_name*

DESCRIPTION

The **ifstat** command displays statistics about packets received and sent on a specified network interface or on all network interfaces. The statistics are cumulative since the filer was booted.

The **-z** argument clears the statistics. The **-a** argument displays statistics for all network interfaces including the virtual host and the loopback address. If you don't use the **-a** argument, specify the name of a network interface.

EXAMPLES

The following command displays network statistics for an Ethernet interface named **e7**:

ifstat e7

The following command displays network statistics for the loopback address:

ifstat lo

The following command displays network statistics for all network interfaces on the filer:

ifstat -a

SEE ALSO

ifconfig

NAME

license - license Data ONTAP 5.3 services

SYNOPSIS

license [*service=code*] ...

DESCRIPTION

The **license** command enables you to enter license codes for specific Data ONTAP 5.3 services. The license codes are provided by Dell. With no arguments, the **license** command prints the current list of licensed services and their codes. It also shows the services that are not licensed for your filer.

The filer is shipped with license codes for all purchased services, so you need to enter the **license** command only after you purchase a new service or after you reinstall the file system.

To disable a license, enter the code **DISABLE**.

All license codes are case-insensitive. Do not leave a space before or after the equal sign in the command.

The following list describes the services you can license:

Enter **nfs** to enable NFS.

Enter **cifs** to enable CIFS.

Enter **http** to enable HTTP.

Enter **snapmirror** to enable SnapMirror.

Enter **snaprestore** to enable SnapRestore.

EXAMPLES

The following example enables NFS:

```
filer> license nfs=ABCDEFG

  nfs license enabled.
      nfs enabled.
```

The following example disables CIFS:

```
filer> license cifs=DISABLE

  unlicense cifs.
      cifs will be disabled upon reboot.
```

NAME

logout - use control-D to logout

DESCRIPTION

The filer doesn't have a logout command. (Since the telnet connection and the console are multiplexed into the same session, there would be no way for a logout command to tell which connection to drop.) To log out, type control-D.

Over telnet, typing control-D disconnects the session.

On the console, typing control-D returns the console to the password prompt. If no password is set, control-D has no effect.

SEE ALSO

passwd

NAME

maxfiles - increase the number of files the volume can hold

SYNOPSIS

maxfiles [*vol_name* [*max*]]

DESCRIPTION

maxfiles increases the number of files that a volume can hold to *max*. Once increased, the value of *max* can never be lowered, so the new value must be larger than the current value. If no argument is specified, **maxfiles** displays the current value of *max* for all volumes in the system. If just the *vol_name* argument is given, the current value of *max* for the specified volume is displayed.

Because each allowable file consumes disk space, and because the value of *max* can never be reduced, increasing *max* consumes disk space permanently. If **maxfiles** identifies a new size as unreasonably large, it will query the user to verify that the new value is correct.

The filer's **df** command (see **df**) can be used to determine how many files have currently been created in the file system.

SEE ALSO

df

NAME

mt - magnetic tape positioning and control

SYNOPSIS

mt [**-f** | **-t** *tapedevice*] *command* [*count*] [*command* [*count*] ...]

DESCRIPTION

mt is used to position or control the specified magnetic tape drive supporting the commands listed below. Commands that support a *count* field allow multiple operations to be performed (the rewind, status and offline commands do not support a count field). **mt** will output failure messages if the specified tape drive cannot be opened or if the operation fails.

The **-f** option specifies which tape device to use. Use **sysconfig** **-t** to list all tape devices on the filer. **-t** has the same effect as **-f**.

USAGE

eof, **weof** Writes *count* end-of-filemarks beginning at the current position on tape.

fsf Forward spaces over *count* filemarks. Positions the tape on the end-of-tape side of the filemark.

bsf Backward spaces over *count* filemarks. Positions the tape on the beginning-of-tape side of the filemark.

fsr Forward spaces *count* records. Positions the tape on the end-of-tape side of the record(s).

bsr Backward spaces *count* records. Positions the tape on the beginning-of-tape side of the record(s).

erase Erases the tape beginning at the current tape position. When the erase completes, the tape is positioned to beginning-of-tape.

rewind Rewinds the tape, positioning the tape to beginning-of-tape.

status Displays status information about the tape unit.

offline Rewinds the tape and unloads tape media.

diag Enables or disables display of diagnostic messages from tape driver. Enabling diagnostic messages can be helpful when attempting to diagnose a problem with a tape device. Specifying a count of "1" enables display of diagnostic messages, a count of "0" disables diagnostic messages. Diagnostic messages are *dis*a*bled* by default.

eom Positions the tape to end of data (end of media if tape is full).

EXAMPLES

The following example uses **mt** to display status information for the no-rewind tape device, unit zero, highest format (density):

```
filer> mt -f nrst0a status

    Tape drive: Exabyte 8505 8mm
       Status: ready, write enabled
       Format: EXB-8500C (w/compression)
       fileno = 0  blockno = 0  resid = 0
```

To skip over a previously created dump file to append a dump onto a no-rewind tape device, use the **fsf** (forward space file) command:

```
filer> mt -f nrst0a fsf 1
```

SEE ALSO

sysconfig, tape

NAME

netstat - show network status

SYNOPSIS

netstat [-**an**]

netstat -**mnrs**

netstat -**i | -I** *interface* [-**dn**]

netstat -**w** *interval* [-**i | -I** *interface*] [-**dn**]

netstat [-**p** *protocol*]

DESCRIPTION

The **netstat** command symbolically displays the contents of various network-related data structures. There are a number of output formats, depending on the options for the information presented. The first form of the command displays a list of active sockets for each protocol.

The second form presents the contents of one of the other network data structures according to the option selected.

The third form will display cumulative statistics for all interfaces or, with an *interface* specified using the **-I** option, cumulative statistics for that interface. It will also display the sum of the cumulative statistics for all configured network interfaces.

The fourth form continuously displays information regarding packet traffic on the interface that was configured first, or with an *interface* specified using the **-I** option, packet traffic for that interface. It will also display the sum of the cumulative traffic information for all configured network interfaces.

The fifth form displays statistics about the protocol specified by *protocol*.

OPTIONS

-**a**	Show the state of all sockets; normally sockets used by server processes are not shown.
-**d**	With either interface display (option **-i** or an interval, as described below), show the number of dropped packets.
-**I** *interface*	Show information only about this interface. When used in the third form with an *interval* specified as described below, information about the indicated *interface* is highlighted in a separated column. (The default interface highlighted is the first interface configured into the system.)
-**i**	Show the state of interfaces that have been configured.
-**m**	Show statistics recorded by the memory management routines for the network's private pool of buffers.

-**n** Show network addresses as numbers. **netstat** normally interprets addresses and attempts to display them symbolically. This option may be used with any of the display formats that display network addresses.

-**p** *protocol* Show statistics about *protocol* , which is one of **tcp**, **udp**, **ip**, or **icmp**. A null response typically means that there are no interesting numbers to report. The program will complain if *protocol* is unknown or if there is no statistics routine for it.

-**s** Show per-protocol statistics. If this option is repeated, counters with a value of zero are suppressed.

-**r** Show the routing tables. When -**s** is also present, show routing statistics instead.

-**w** *wait* Show network interface statistics at intervals of *wait* seconds.

DISPLAYS

The default display, for TCP sockets, shows the local and remote addresses, send window and send queue size (in bytes), receive window and receive queue sizes (in bytes), and the state of the connection. For UDP sockets, it shows the local and remote addresses, and the send and receive queue size (in bytes). Address formats are of the form "host.port" or "network.port" if a socket's address specifies a network but no specific host address. If known, the host and network addresses are displayed symbolically according to the databases **/etc/hosts** and **/etc/networks**, respectively. If a symbolic name for an address is not known, or if the -**n** option is specified, the address is printed numerically, according to the address family. Unspecified, or "wildcard", addresses and ports appear as "*".

The interface display specified by the -**i** or -**I** options provides a table of cumulative statistics regarding packets transferred, errors, and collisions. The network addresses of the interface and the maximum transmission unit ("mtu") are also displayed. If the interface is currently down, then a "*" is appended to the interface name.

When an *interval* is specified, a summary of the interface information, consisting of packets transferred, errors, and collisions, is displayed.

The routing table display indicates the available routes and their status. Each route consists of a destination host or network and a gateway to use in forwarding packets. The flags field shows a collection of information about the route stored as binary choices; the flags are:

2 Protocol-specific routing flag #2 (for ARP entries, means that the entry is "published").

C Use of this route will cause a new route to be generated and used.

D The route was created dynamically by a redirect.

G The route is to a gateway.

H The route is to a host (otherwise, it's to a net).

L The route includes valid protocol to link address translation.

M The route was modified dynamically by a redirect.

R The route has timed out.

S The route was manually added with a **route** command (see **route**).

U The route is usable ("up").

Direct routes are created for each interface attached to the local host; the gateway field for such entries shows the address of the outgoing interface. The refcnt field gives the current number of active uses of the route. Connection oriented protocols normally hold on to a single route for the duration of a connection while connectionless protocols obtain a route whenever they transmit to a destination. The use field provides a count of the number of packets sent using that route. The interface entry indicates the network interface utilized for the route.

When **netstat** is invoked with the **-w** option and an *interval* argument, it displays a running count of statistics related to network interfaces. An obsolescent version of this option used a numeric parameter with no option, and is currently supported for backward compatibility. This display consists of a column for the primary interface and a column summarizing information for all interfaces. The default primary interface is the first interface configured into the system. The primary interface may be replaced with another interface with the **-I** option. The first line of each screen of information contains a summary since the system was last rebooted. Subsequent lines of output show values accumulated over the preceding interval.

FILES

/etc/hosts	host name database
/etc/networks	network name database

SEE ALSO

ifconfig, nfsstat, sysstat, hosts, networks

NAME

nfs - turn NFS service off and on

SYNOPSIS

nfs [**on** | **off**]

DESCRIPTION

nfs turns NFS service off or on. With no arguments, **nfs** shows the current state of NFS service. **nfs** is normally used in the initialization command script, **/etc/rc**.

FILES

/etc/rc system initialization command script

SEE ALSO

rc

NAME

nfsstat - display NFS statistics

SYNOPSIS

nfsstat [*interval*]

nfsstat [-**c**] [-**t**] -**h** [*ip_address* | *host_name*]

nfsstat -**l** [-**t**]

nfsstat -**z**

DESCRIPTION

nfsstat displays statistical information about NFS (Network File System) and RPC (Remote Procedure Call) for the filer. It can also be used to reinitialize this information. If no arguments are given, **nfsstat** displays statistical information since last zeroed with the -**z** option (or since reboot if statistics have not been zeroed).

If the *interval* argument is specified, **nfsstat** continuously displays the summary information for the following NFS requests: getattr, lookup, readlink, read, write, create, remove, and readdir/readdirplus. The first line of data displayed, and every 20th line thereafter, contains cumulative statistics. Each subsequent line shows incremental statistics for the *interval* (in seconds) since the last display.

Per-client statistics can also be collected and displayed by enabling the **nfs.per_client_stats.enable** options (using the **options** command - see **options**) and invoking **nfsstat** with the -**h** or the -**l** options. Per-client statistics are collected for up to the first 256 NFS clients that have mounted the filesystem on the given filer.

OPTIONS

-**h** Displays per-client statistics since last zeroed with the -**z** option (or since reboot if statistics have not been zeroed). The statistics are displayed on a per-client basis, with the IP address and host name (where available) of each client being displayed at the head of each client's block of statistics.

If an optional IP address or host name is specified with the -**h** option, only the statistics associated with this client are displayed.

-**l** Displays a list of the clients whose statistics have been collected on a per-client basis, along with the total NFS calls for that client since last reboot, or last zeroed with the -**z** option, the count being displayed both as the actual count and as a percentage of calls from all clients.

-**z** Zeroes (reinitializes) the current statistics. (However, statistics since boot are also retained.)

-**c** Includes reply cache statistics in the data displayed.

-**t** Displays the statistics since boot time, rather than since the last time they were zeroed.

DISPLAYS

The server RPC display includes the following fields, with separate values for TCP and UDP:

calls The total number of RPC calls received.

badcalls The total number of calls rejected by the RPC layer (the sum of **badlen** and **xdrcall** as defined below).

nullrecv The number of times an RPC call was not available when it was thought to be received.

badlen The number of RPC calls with a length shorter than a minimum-sized RPC call.

xdrcall The number of RPC calls whose header could not be XDR decoded.

The server NFS display shows the number of NFS calls received (**calls**) and rejected (**badcalls**), and the counts and percentages for the various calls that were made.

SEE ALSO

netstat, options, sysstat

NAME

options - display or set filer options

SYNOPSIS

options [*option value*] ...

DESCRIPTION

options is used to change configurable filer software options. If no options are specified, then **options** prints the current value of all available options. The default *value* for most options is **off**, which means that the option is not set. Changing the *value* to **on** enables the option; for most options, the only valid values are **on** (which can also be expressed as **yes**, **true**, or **1**) in any mixture of upper and lower case, and **off** (which can also be expressed as **no**, **false**, or **0**) in any mixture of upper and lower case. The description of the option will indicate the default if it is not **off**, and will indicate what values are allowed if it isn't an on/off option. If it is desired to make an option setting permanent, the necessary **options** command must be placed in the **/etc/rc** file, as options settings are not preserved across system reboots. The legal options are as follows:

cifs.access_logging_enable
When on, enables the filer to process access logging, or auditing, information. The default is off.

cifs.access_logging.filename
Specifies the active event log file. The file must be in an existing directory in a network share.

cifs.bypass_traverse_checking
When on (the default), directories in the path to a file are not required to have the 'X' (traverse) permission. This option does not apply in UNIX qtrees.

cifs.guest_account
Enables a user to get access to the filer provided that either the filer uses a Domain Controller for authentication and the user is not in a trusted domain, or the filer uses the **/etc/passwd** file or the NIS password database for authentication and the user has no entry in the **/etc/passwd** or the NIS password database. If this option is set to the name of an account in the password database, a user logging into the filer will be assigned to the guest account if their name is not listed in the password database (when using **/etc/passwd** or NIS) or if the user is not from a trusted domain (when using a domain controller). The configured user name will be used for the UNIX user ID, group ID, and group set of the specified account. If the option is blank, guest access is disabled.

cifs.home_dir When set to the pathname of a directory, this defines the path to the "homes directory". The directories under this path should have the names of users as their names. When a user connects to the filer using CIFS and there is a directory name that exactly matches

the user's lower-cased Windows login name, they will see a share of that name (truncated to 12 characters) that is their "home directory". Only the user can access the home directory using this share. All other users cannot see the share name since they are logged in under a different user.

cifs.idle_timeout

Specifies the amount of idle time in seconds before the filer disconnects a session. An idle session is a session in which a user does not have any files opened on the filer. The value of this option ranges from 600 to 4,000,000 (effectively infinite). The default is 1800.

cifs.netbios_aliases

Provides a comma-separated list of alternative names for the filer. A user can connect to the filer using any of the listed names.

cifs.oplocks.enable

When **cifs.oplocks.enable** is on (the default), the filer allows clients to use oplocks (opportunistic locks) on files. Oplocks are 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, this option should be disabled only to isolate problems.

cifs.perm_check_use_gid

This option affects security checking for Windows clients of files with UNIX security where the requestor is not the file owner. In all cases Windows client requests are checked against the share-level ACL, then if the requestor is owner, the "user" perms are used to determine the access.

If the requestor is not owner and if perm_check_use_gid is "on" it means files with UNIX security are checked using normal UNIX rules, i.e. if the requestor is a member of the file's owning group the "group" perms are used, otherwise the "other" perms are used.

If the requestor is not owner and if perm_check_use_gid is "off", files with UNIX security style are checked in a way which works better when controlling access via share-level ACLs. In that case the requestor's desired access is checked against the file's "group" permissions, and the "other" permissions are ignored. In effect, the "group" perms are used as if the Windows client were always a member of the file's owning group, and the "other" perms are never used.

The default setting is "on" for new installations. For existing installations, this has the opposite effect of the old "PC-mode" installation setting.

If you do not plan to use share-level ACLs to control access to UNIX security style files (e.g. in a UNIX qtree), you might wish to change this setting to "on."

cifs.save_case

By default, the filer preserves the case of CIFS names, even though a case insensitive hash and search is done. Setting this option to OFF forces names to be saved in lower case, avoiding some of the case conversion problems with non-ASCII characters and preventing PC applications from changing the names of files that were created with lower case names from UNIX.

cifs.scopeid NetBIOS scope IDs allow the system administrator to create small workgroups out of a network by partitioning the NetBIOS name space; only clients with the same NetBIOS scope ID as the filer will be able to use the filer as a CIFS server. Normally, 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 scope ID of that scope. The scope ID can be changed only when CIFS is not running.

cifs.search_domains

Specifies a list of domains that trust each other to search for a mapped account. The argument for the option is a comma-separated list that is searched in order. If no list is supplied, all domains are searched. You use this option to control searches if you used an asterisk for a domain name in the usermap.cfg file.

cifs.show_snapshot

By default this option is FALSE. The snapshot directory ~snapshot is no longer shown at the root of a share. This is a change in behavior from previous versions. Setting this to TRUE will restore the old behavior. On Windows NT 4 or Windows 95 clients, the user can access snapshots by entering \\filer\share\.snapshot (or ~snapshot or ~snapsht) in the Start->Run menu. Snapshots can also be accessed lower in the share by providing a path to a lower directory. Snapshots can be accessed through DOS on any system by changing to the ~snapsht directory.

NOTE: When this option is TRUE it can confuse programs like FastFind that don't know about snapshots.

cifs.symlinks.cycleguard

The **cifs.symlinks.cycleguard** option (on by default), eliminates the possibility of traversing directories cyclically during the process of following symbolic links. With this option set to on, if the target of the symlink resolves to a directory that is directly above the symlink's parent directory, it is disallowed.

With this option set to off, many standard Windows apps (such as Find in Win95 / NT4.0) will not operate correctly when a symlink points to a parent directory. This is because they do not understand

symbolic links and will repeatedly loop on them. Users should use caution when changing this option.

cifs.symlinks.enable

When **cifs.symlinks.enable** is on (the default), if the object being accessed by a CIFS client is a symbolic link (whether absolute or relative), the filer follows the link with the proviso that the ultimate target turns out to reside within the originating share (thus ensuring that the client has access permission to the target).

cifs.trace_login

When **cifs.trace_login** is on (the default is off), the filer logs all login-related activities. This can be used to diagnose access problems on the filer.

console.encoding

Specifies how non-ASCII character information is presented. The value can be:

nfs - NFS character set. You can use both NFS extended (> 0x7F) and SGML characters for input.

sgml - SGML character format. You can use both NFS extended (greater than 0x7F) and SGML characters for input.

utf8 - UTF-8 character sets. For input, any character greater than 0x7F is the beginning of a UTF-8 encoding.

The default is nfs.

dns.domainname

Sets the DNS domainname to the specified domainname.

dns.enable Enables DNS client on the filer. The DNS domain must be set and the **/etc/resolv.conf** file must exist prior to enabling DNS.

ip.path_mtu_discovery.enable

Enables/disables path MTU discovery; it is currently used only by TCP. Path MTU discovery allows a host to discover the "maximum transmission unit", i.e. the largest link-level packet that can be transmitted, over a path from that host to another host. This means that the filer needn't choose a conservative packet size for a TCP connection to a host not on the same net as the filer, but can attempt to discover the largest packet size that can make it to the other host without fragmentation.

httpd.enable

Enables HTTP access to the filer.

httpd.admin.enable

Enables HTTP access to the administration area of the filer, via a private URL: any URL beginning with **/na_admin** is mapped to the directory **/etc/http**. Thus, a man page on the filer *filer* with the file

name **/etc/http/man/***name* can be accessed with the URL **http://***filer***/na_admin/man/***name*.

httpd.log.max_file_size

Specifies the maximum size that the HTTP log file **/etc/log/httpd.log** can grow to. The default is 2147483647, which is the largest file size that many clients support.

httpd.rootdir

Specifies the complete pathname of the root directory that contains files and subdirectories for HTTP access.

httpd.timeout

Specifies the minimum amount of time (in seconds) before an idle HTTP 15 connection will time out. The default is 900 seconds, which is fifteen minutes.

httpd.timewait.enable

When enabled, the filer will put HTTP connections that have been closed by the client into the TIME_WAIT state for one minute, which is twice the maximum segment lifetime (2*MSL). By default, TIME_WAIT state is bypassed for HTTP connections.

ip.match_any_ifaddr

If the option is on, the filer will accept any packet that is addressed to it even if that packet came in on the wrong interface. If you are concerned about security, you should turn this off.

nfs.mount_rootonly

When enabled, the mount server will deny the request if the client is not root user using privileged ports. By default, the feature is enabled for more secure access.

nfs.per_client_stats.enable

Enables/disables the collection and display of per-client NFS statistics, as described in **nfsstat**.

nfs.tcp.enable When enabled, the NFS server supports NFS over TCP. By default, the feature is enabled; it can be disabled if there is a problem with some client when using NFS over TCP, and that client cannot be configured to use NFS over UDP.

nfs.v2.df_2gb_lim

Causes the filer to return replies to the "file system statistics" NFS version 2 request that show no more than (2**31)-1 (or 2,147,483,647) total, free, or available bytes (i.e., 2GB) on the file system.

Some NFS clients require this option because, if they get return values from the "file system statistics" request with more than the specified number of bytes, they'll incorrectly compute the amount of free space on the file system, and may think that there's no free space on a file system that has more than 2GB free.

nfs.v3.enable
When enabled, the NFS server supports NFS version 3. By default, the feature is enabled; it can be disabled if there is a problem with some client when using NFS version 3, and that client cannot be configured to use NFS version 2.

nfs.webnfs.enable
When enabled, the NFS server supports WebNFS lookups. By default, WebNFS lookups are disabled.

nfs.webnfs.rootdir
Specifies the WebNFS rootdir. Once the rootdir is set, WebNFS clients can issue lookups relative to the rootdir using the public filehandle.

nfs.webnfs.rootdir.set
After specifying the rootdir, this option needs to be enabled for the rootdir setting to take effect. Disabling this option disables the existing rootdir setting.

nis.domainname
Sets the NIS domain to the specified domainname.

nis.enable Enables NIS client on the filer. The NIS domain must be set prior to enabling NIS.

raid.timeout
Sets the time in hours, as a number greater than or equal to **1**, that the system will run after a single disk failure has caused the system to go into *degraded mode*. The default is **24**. If the **raid.timeout** option is specified after the system is already in *degraded mode*, the timeout is set to the value specified and the timeout restarted.

raid.reconstruct_speed
Specifies the speed at which the RAID reconstruction should occur ranging from the slowest speed **1** to the fastest speed possible **10**. The RAID reconstruction process is given more cpu time as the speed is increased, so increasing the speed of the reconstruction will take away cpu time for network operations. The default speed is **4**, which is roughly 40% of the cpu time, though more time may be used if there is idle time available.

raid.scrub.enable
Enables/disables the RAID scrub feature (see **disk**). By default, it is enabled. This option only affects the scrubbing process that gets started from cron. For user requested scrubs, this option is ignored.

rsh.enable Enables the RSH server on the filer.

snmp.enable
Enables the SNMP server on the filer.

telnet.enable

Enables the Telnet server on the filer.

telnet.hosts

Specifies up to 5 clients that will be allowed telnet access to the server. The host names should be entered as a commaseparated list with no spaces in between. Enter a "*" to allow access to all clients; this is the default. Enter a "-" to disable telnet access to the server.

timed.enable

Determines whether a time daemon (**timed**) runs on the filer. If **timed.enable** is on, the filer synchronizes its time with a time server.

timed.log Specifies whether time changes initiated by timed should be logged to the console.

timed.max_skew

Specifies the maximum amount of skew between the time reported by the time server and the filer's time that we will allow when synchronizing the time. If the difference in the time reported by the server and the filer's time is greater than this value, the filer will not synchronize to the time reported by the time server. The maximum skew is specified in seconds (suffix s), minutes (suffix m), or hours (suffix h). Defaults to "30m".

timed.proto

Specifies the protocol used to synchronize time. "rdate" specifies the "rdate;" "sntp" specifies the Simple Network Time Protocol.

timed.sched

Specifies the timed synchronization schedule. There are several pre-defined schedules:

hourly synchronize every hour (the default)

multihourly synchronize every 6 hours

daily synchronize every day at midnight

Custom schedules may also be specified by giving the number of minutes or hours between time synchronization. Minutes are specified by digits followed by an "m"; hours are specified by digits followed by an "h". For example, *options timed.sched 2h* will cause time to be synchronized every two hours.

To avoid overburdening the time server, the filer randomly selects the exact time of the synchronization within a 20-minute window.

timed.servers

Specifies up to five time servers used by the time daemon. Time servers are contacted in the order specified; if a server can't be contacted, the time daemon tries the next one in the list.

wafl.convert_ucode

Setting this option to ON forces conversion of all directories to Unicode format when accessed from both NFS and CIFS. By default (OFF), access from CIFS causes conversion of pre-4.0 and 4.0 format directories; access from NFS causes conversion of 4.0 format directories.

vol.copy.throttle

Specifies the default speed of all volume copy operations. The speed can be a number in the range from 1 to 10, 10 being the highest speed and the default.

wafl.create_ucode

Setting this option to ON forces Unicode format directories to be created by default, both from NFS and CIFS. By default(OFF), all directories are created in pre-4.0 format and the first CIFS access will convert it to Unicode format.

wafl.default_nt_user

Specifies the NT user account to use when a UNIX user accesses a file with NT security (has an ACL), and that UNIX user would not otherwise be mapped. If this option is set blank, such accesses will be denied.

wafl.default_UNIX_user

Specifies the UNIX user account to use when an NT user attempts to log in and that NT user would not otherwise be mapped. If this option is set blank, such accesses will be denied.

wafl.maxdirsize

Sets the maximum size (in K-Bytes) that a directory can grow to. This is set to 10240 by default; it limits directory size to 10MBytes and can hold over 300,000 files. Most users should not need to change this setting. This option is useful for environments where system users may grow a directory to a size that starts impacting system performance. When a user tries to create a file in a directory that is at the limit, the system returns a ENOSPC error and fails the create.

wafl.nt_admin_priv_map_to_root

When on (the default), an NT administrator is mapped to UNIX root.

wafl.root_only_chown

When enabled, only the root user can change the owner of a file. When disabled, non-root users can change the owner of files that they own. When a non-root user changes the owner of a file they own, both the set-UID and set-GID bits of that file are cleared for security reasons. A non-root user is not allowed to give away a file if it would make the recipient overrun its user quota.
wafl.root_only_chown is enabled by default.

wafl.wcc_minutes_valid

Specifies the number of minutes a WAFL credential cache entry is valid. The value can range from 1 through 20160. The default is 20.

Multiple options can be set at once in an **options** command. For example:

options nfs.tcp.enable on nfs.v2.df_2gb_lim on raid.timeout 48

sets **nfs.tcp.enable** to **on**, sets **nfs.v2.df_2gb_lim** to **on**, and sets **raid.timeout** to **48**.

SEE ALSO

disk, nfsstat, snap, autosupport

NAME

passwd - modify the system administrative user's password

SYNOPSIS

passwd

DESCRIPTION

passwd changes the filer's administrative user's password. First it prompts you for the login name (if any non-root users are configured). Then it prompts you for the current password. If you type the current password correctly, the filer requests a new password. The **passwd** command imposes no minimum length or special character requirements for " **root** ". As with any password, it is best to choose a password unlikely to be guessed by an intruder. All non-root administrative user's password should meet the following restrictions:

- it should be at least 6 characters long
- it should contain at least two alphabets
- it should contain at least one digit or special character

If the filer is booted from floppy disk, selection "(3) Change password" enables you to reset the "root" password without entering the old password.

SEE ALSO

useradmin

NAME

ping - send ICMP ECHO_REQUEST packets to network hosts

SYNOPSIS

ping [-**s**] [-**Rrv**] *host* [*packetsize* [*count*]]

DESCRIPTION

ping uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from the specified *host* or gateway. ECHO_REQUEST datagrams have an IP and ICMP header, followed by a *struct timeval* and then an arbitrary number of bytes used to fill out the packet. If *host* responds, **ping** prints "*host* is alive." Otherwise, **ping** will resend the ECHO_REQUEST once a second. If the *host* does not respond after *count* seconds (default value is 20), **ping** will print "no answer from *host*."

When the -**s** flag is specified, **ping** sends one datagram per second and prints one line of output for every ECHO_RESPONSE that it receives. **ping** computes the round-trip times and packet loss statistics. When the *count* number of packets have been sent or if the command is terminated with a ^C, the summary statistics is displayed. The default *packetsize* is 56, which translates into 64 ICMP bytes when combined with the 8 bytes of ICMP header.

OPTIONS

-**R** Record route. Includes the RECORD_ROUTE option in the ECHO_REQUEST packet and displays the route buffer on returned packets. Note that the IP header is only large enough for nine such routes. Many hosts ignore or discard this option.

-**r** Bypass the normal routing tables and send directly to a host on an attached network. If the host is not on a directly-attached network, an error is returned.

-**s** Send one datagram every second.

-**v** Verbose output. ICMP packets other than ECHO_RESPONSE that are received are listed.

SEE ALSO

ifconfig, netstat

NAME

qtree - create and manage qtrees

SYNOPSIS

qtree

qtree create [*name*]

qtree security [*name* [**UNIX** | **ntfs** | **mixed**]]

qtree oplocks [*name* [**enable** | **disable**]]

DESCRIPTION

The **qtree** command creates qtrees and specifies attributes for qtrees.

A qtree can be an entire volume or a subset of a volume. It is similar to a partition in that you cannot move files into or out of a qtree. There are, however, two differences between a qtree and a partition:

A qtree is more flexible than a partition because you can change the size of a qtree at any time.

A qtree enables you to apply attributes such as oplocks and security style to a subset of files and directories rather than to an entire volume.

If there are files and directories in a volume that do not belong to any qtrees you create, the filer considers them to be in qtree 0. Qtree 0 can take on the same types of attributes as any other qtrees.

You can use any **qtree** command whether or not quotas are enabled on your filer.

The **qtree** command without any arguments displays the attributes of all quota trees on the filer.

The **qtree create** command creates a qtree. It is equivalent to the **quota qtree** command. If *name* does not begin with a slash (/), the qtree is created in the root volume. To create a qtree in a particular volume, specify *name* in this format: /vol/*vol_name*/*qtree_name*.

A qtree can be created only in the root directory of a volume. By default, a qtree has the same security style as the root directory of the volume and oplocks are enabled. The root directory of a volume, by default, uses the UNIX security style.

A qtree does not have any restrictions on disk space or the number of files. To impose these restrictions on a qtree, edit the /etc/quotas file. Refer to the **quotas** man page for more information about the file format. To make the changes to the /etc/quotas file go into effect, use the **quota** command. Refer to the **quota** man page for more information about the **quota** command.

If you enter the **qtree create** command without arguments, the command displays all existing qtrees and their attributes.

To delete a qtree, remove it from a client as you would any directory. You can create up to 254 qtrees on a filer.

The **qtree security** command changes the security style for files and directories. Security style means the method the filer uses to determine whether a user has access to a file. If *name* is the path name to a qtree, the security style applies to the files and directories in the specified qtree. The path name to a qtree does not need to end with a slash. If *name* is a path name to a volume, the security style applies to those directories and files in qtree 0. Any new qtree you create inherits the security style from qtree 0 by default. The path name to a volume must end with a slash.

The security style can be one of the following values:

UNIX — The user's UID and GID, and the UNIX-style permission bits of the file or directory determine user access. The filer uses the same method for determining access for both NFS and CIFS requests. If you change the security style of a qtree or a volume from ntfs to UNIX, the filer disregards the Windows NT permissions that were established when the qtree or volume used the ntfs security style.

ntfs — For CIFS requests, Windows NT permissions determine user access. For NFS requests, the filer generates and stores a set of UNIX-style permission bits that are at least as restrictive as the Windows NT permissions. The filer grants NFS access only if the UNIX-style permission bits allow the user access.

If you change the security style of a qtree or a volume from UNIX to ntfs, files created before the change do not have Windows NT permissions. For these files, the filer uses only the UNIX-style permission bits to determine access.

mixed — Some files in the qtree or volume have the UNIX security style, and some have the ntfs security style. A file's security style depends on whether the permission was last set from CIFS or NFS. For example, if a file currently uses the UNIX security style and a CIFS user sends a set-ACL request to the file, the file's security style is changed to ntfs. If a file currently uses the ntfs style and an NFS user sends a set-permission request to the file, the file's security style is changed to UNIX.

If you do not specify UNIX, ntfs, or mixed in the **qtree security** command, the security style for *name* is displayed. If you omit *name*, the security styles for all qtrees on the filer are displayed.

The **qtree oplocks** command enables or disables oplocks for files and directories in a qtree or in a volume. If *name* is the path name to a qtree, the attribute applies to files and directories in the specified qtree. The path name to a quota tree does not need to end with a slash. If *name* is the path name to a volume, the attribute applies to those files and directories in qtree 0. The path name to a volume must end with a slash.

If the **cifs.oplocks.enable** option is off, oplocks are not sent even if you enable the oplocks on a per-quota-tree basis with the **qtree oplocks** command. The **cifs.oplocks.enable** option is enabled by default.

If you do not specify enable or disable in the **qtree oplocks** command, the oplock attribute for *name* is displayed. If you omit *name*, the oplock attributes for all quota trees on the filer are displayed.

EXAMPLES

The following example sets the security style of a qtree named marketing in the root volume to ntfs:

filer> **qtree security marketing ntfs**

The following example sets the security style of a qtree named engineering in the vol1 volume to ntfs:

filer> **qtree security /vol/vol1/engr ntfs**

The following example sets the security style of the root volume to UNIX:

filer> **qtree security / UNIX**

The following example sets the security style of the vol1 volume to UNIX: filer> **qtree security /vol/vol1/ UNIX**

The following example disables oplocks for the engr qtree:

filer> **qtree oplocks /vol/vol1/engr disable**

The following example enables oplocks for the vol1 volume:

filer> **qtree oplocks /vol/vol1/ disable**

The following example displays the security and oplocks attributes for all volumes and qtrees on the filer:

filer> **qtree**

Volume	Tree	Style	Oplocks
vol0		UNIX	enabled
vol0	marketing	ntfs	enabled
vol1		UNIX	enabled
vol1	engr	ntfs	disabled

SEE ALSO

options, quota, quotas

NAME

quota - control filer disk quotas

SYNOPSIS

quota [**on** | **off** | **resize**] [*volume*]

quota report [*path*]

quota qtree [*name*]

DESCRIPTION

A quota limits the amount of disk space and the number of files that a particular user or group can consume. A quota can also restrict the total space and files used in a qtree, or the usage of users and groups within a qtree. A request that would cause a user or group to exceed an applicable quota fails with a "disk quota exceeded" error. A request that would cause the number of blocks or files in a qtree to exceed the qtree's limit fails with an "out of disk space" error.

User and group quotas do not apply to the root user; tree quotas, however, do apply even to root.

The **quota** command controls quotas, and the /etc/quotas file describes the quotas to impose. All quotas are established on a per-volume basis. For further information on the format of the /etc/quotas file, refer to the **quotas** man page.

With no arguments, the **quota** command indicates whether quotas are on or off in each volume. The following list describes how to use the various **quota** commands:

quota on *volume*

activates quotas in the specified volume based on the contents of **/etc/quotas**. The volume name may be omitted if the system has only one volume. Changing **/etc/quotas** has no effect until the next time **quota on** or **quota resize** is executed. The filer remembers whether quotas are on or off even after a reboot, so **quota on** should *not* be added to **/etc/rc**. When quotas are first turned on, the filer scans the file system to determine current file and space usage for each user and group with a quota. This may take several minutes during which quotas are not in effect, although the file system is still accessible. Executing **quota** with no arguments during this period indicates that quotas are initializing and reports how much of the initialization process has completed.

quota off *volume*

turns quotas off on the specified volume. The volume name may be omitted if the system has only one volume.

quota resize *volume*

adjusts currently active quotas in the specified volume to reflect changes in the **/etc/quotas** file. For instance, if you edit

an entry in **/etc/quotas** to increase a user's quota, **quota resize** will cause the change to take effect. The volume name may be omitted if the system has only one volume. **quota resize** can be used only when quotas are already on. Because it does not rescan the file system to compute usage, **quota resize** is faster than turning quotas off and then on again. **quota resize** will apply all updated entries in **/etc/quotas**; however, it will generally ignore newly added entries. A newly added entry will only take effect if the corresponding user or group has an active quota as a result of updating a file subject to default quotas.

quota report prints the current file and space consumption for each user or group with a quota and for each qtree. With a *path* argument, **quota report** displays information about all quotas that apply to the file.

FILES

/etc/quotas quota configuration file

SEE ALSO

rc, rquotad, qtree

DIAGNOSTICS

If **/etc/quotas** is incorrectly formatted, or if a specified file doesn't exist, then **quota on** prints a warning and does not turn quotas on.

NAME

rdate - set system date from a remote host

SYNOPSIS

rdate *hostname*

DESCRIPTION

rdate sends a request to the time server on *hostname* and sets the local date and time to the value returned by the server. **rdate** will time out if the server doesn't respond in 10 seconds.

rdate can be added to **/etc/rc** to automatically synchronize the system time with the time server on each reboot.

FILES

/etc/rc system initialization command script

SEE ALSO

date, rc

NAME

reboot - stop and then restart the filer

SYNOPSIS

reboot [-**d**] [-**t** *minutes*]

DESCRIPTION

reboot halts the filer and then restarts it. **reboot** is commonly used to allow modified configuration files to take effect or to run a newly installed version of Data ONTAP 5.3.

NFS clients can maintain use of a file over a **halt** or **reboot** (although experiencing a failure to respond during that time), but CIFS clients cannot do so safely. Therefore CIFS clients should -if possible- be warned to close their open files. If you did not use the -**t** option to specify a maximum delay *and* there are CIFS clients with open files, the **reboot** command displays the number of CIFS users and the number of open CIFS files. Then it prompts you for the number of minutes to delay. CIFS files that are still open at the time the filer halts will lose writes that had been cached but not written.

reboot logs a message in the **/etc/messages** file (see **messages**) file to indicate that the filer was rebooted on purpose.

OPTIONS

-**d** Dump system core before rebooting.

-**t** *minutes* Reboots after the indicated number of minutes, or after all CIFS files that were open have been closed, whichever is sooner.

SEE ALSO

download, halt, savecore, setup

NAME

restore - restore files or file systems from backups made with the filer's dump command

SYNOPSIS

restore *key args...*

DESCRIPTION

The **restore** restores files from backup tapes created with the **dump** (see **dump**) command. A full backup of a file system may be restored and subsequent incremental backups layered on top of it. The actions of **restore** are controlled by the given *key*, which is a string of characters containing at most one function letter and possibly one or more function modifiers.

The function portion of the key is specified by one of the following letters:

r Restores (rebuilds a file system or subtree). The target subtree should be made pristine by removing it from a client of the server or, if the entire file system or all subtrees of the file system are to be restored, by booting from floppy disk and selecting the "Install new file system." option, before starting the restoration of the initial level 0 backup. If the level 0 restores successfully, the **r** key may be used to restore any necessary incremental backups on top of the level 0.

Note that **restore r** will restore *all* files from the dump tape(s).

An example:

restore rf rst0a

Note that **restore** leaves a file **restore_symboltable** in the directory that was dumped to pass information between incremental restore passes. This file should be removed when the last incremental has been restored.

R **restore** requests a particular tape of a multi-volume set on which to restart a full restore (see the **r** key above). This is useful if the restore has been interrupted.

t Lists the names of the specified files if they occur on the backup. If no file argument is given, then the root directory is listed, which results in the entire content of the backup being listed.

x Extracts the named files. If a named file matches a directory whose contents were backed up, the directory is recursively extracted. The owner, modification time, and mode are restored. If no *filename* argument is specified, the backup root directory is extracted. This results in the entire backup being restored.

The following characters may be used in addition to the letter that selects the function desired.

b The next argument to **restore** is used as the block size of the media (in kilobytes). If the **b** option is not specified, **restore** tries to determine the media block size dynamically.

f The next argument to **restore** is used as the name of the archive instead of the standard input. If the name of the file is -, **restore** reads from standard input.

s The next argument to **restore** is a number which selects the file on a multi-file dump tape. File numbering starts at 1.

D By default, files will be restored into the directory from which they were dumped. If the **D** option is specified, the next argument to **restore** is the full absolute pathname of a directory into which the files should be restored.

v Normally **restore** does its work silently. The **v** (verbose) key causes it to type the name of each file it treats preceded by its file type.

y **restore** will not ask whether it should abort the restore if it encounters an error. It will always try to skip over the bad block(s) and continue as best it can.

DIAGNOSTICS

Complains about bad key characters.

Complains if it gets a read error. If **y** has been specified, or the user responds **y**, **restore** will attempt to continue the restore.

If a backup was made using more than one tape volume, **restore** will notify the user when it is time to mount the next volume.

There are numerous consistency checks that can be listed by **restore**. Most checks are self-explanatory or can "never happen". Common errors are given below.

filename: **not found on tape**
The specified file name was listed in the tape directory, but was not found on the tape. This is caused by tape read errors while looking for the file, and from using a dump tape created on an active file system.

expected next file *inumber,* **got** *inumber*
A file that was not listed in the directory showed up. This can occur when using a dump created on an active file system.

Incremental dump too low
When doing incremental restore, a dump that was written before the previous incremental dump, or that has too low an incremental level has been loaded.

Incremental dump too high
When doing incremental restore, a dump that does not begin its coverage where the previous incremental dump left off, or that has too high an incremental level has been loaded.

Tape read error while restoring *filename*
Tape read error while skipping over inode *inumber*
Tape read error while trying to resynchronize
A tape (or other media) read error has occurred. If a file name is specified, then its contents are probably partially wrong. If an inode is being skipped or the tape is trying to resynchronize, then no extracted files have been corrupted, though files may not be found on the tape.

resync restore, skipped *num* **blocks**
After a dump read error, **restore** may have to resynchronize itself. This message lists the number of blocks that were skipped over.

FILES

/tmp/rstdir* file containing directories on the tape.

/tmp/rstmode* owner, mode, and time stamps for directories.

restore_symboltable
information passed between incremental restores.

SEE ALSO

dump

NAME

route - manually manipulate the routing table

SYNOPSIS

route [-**fn**] **add**|**delete** [**host**|**net**] *destination gateway* [*metric*]

DESCRIPTION

route allows the system administrator to manually manipulate the network routing table for the specific host or network specified by *destination*. The *gateway* argument is the nexthop gateway to which packets should be addressed for the corresponding *destination*. The *metric* argument indicates the number of "hops" to the *destination*. The *metric* argument is required for the **add** command; it must be zero if the *destination* is on a directly-attached network, and non-zero if the route is via one or more gateways.

The **add** command adds the specified route for the given *destination* to the routing table. The **delete** command deletes the specified route from the routing table.

Routes to a particular host are distinguished from those to a network by interpreting the Internet address associated with *destination*. The optional keywords **net** and **host** force the destination to be interpreted as a network or a host, respectively. Otherwise, if the *destination* has a "local address part" of INADDR_ANY (i.e., 0), or if the *destination* is the symbolic name of a network, then the route is assumed to be to a network; otherwise, it is presumed to be a route to a host. If the route is to a destination via a gateway, the *metric* parameter should be greater than 0. If *metric* is set to 0, the gateway given is the address of this host on the common network, indicating the interface to be used for transmission.

All symbolic names specified for a *destination* or *gateway* are looked up first as a host name in the **/etc/hosts** database. If this lookup fails, then the name is looked up as a network name in the **/etc/networks** database. "default" is also a valid destination, which is used if there is no specific host or network route.

The netmask for a route to a network is implicitly derived from the class of the network; to override that, the *destination* for a network route can have **/***bits* or **&***mask* after it, where *bits* is the number of high-order bits to be set in the netmask, or *mask* is the netmask (either as a number defaults to decimal, precede with **0x** for hexadecimal, precede with **0** for octal - or as a number IP address).

OPTIONS

-**f** Remove all gateway entries in the routing table. If this is used in conjunction with one of the commands, **route** removes the entries before performing the command.

-**n** Prevent attempts to print host and network names symbolically when reporting actions.

DIAGNOSTICS

add [**host**|**net**] *destination*:*gateway*
The specified route is being added to the table.

delete [**host**|**net**] *destination*:*gateway*
The specified route is being deleted.

destination gateway **done**
When the -**f** flag is specified, each routing table entry deleted is indicated with a message of this form.

network unreachable
An attempt to add a route failed because the gateway listed was not on a directly-connected network. The next-hop gateway must be given.

not in table
A delete operation was attempted for an entry which wasn't present in the table.

entry already exists
An add operation was attempted for an existing route entry.

routing table overflow
An add operation was attempted, but the system was unable to allocate memory to create the new entry.

SEE ALSO

routed

NAME

routed - network routing daemon

SYNOPSIS

routed on

routed off

routed [-**n**] **status**

DESCRIPTION

routed (pronounced "route-D") uses a variant of the Xerox NS Routing Information Protocol (RIP) to manage selection of the default gateway used for IP network routing. The filer's **routed** is different from the standard UNIX **routed** as it never sends RIP packets, or builds route tables from RIP information, but only snoops for RIP exchanges to determine gateway status; it builds the routing table based on ICMP redirects.

When **routed** is started with the **routed on** command, it reads the **/etc/dgateways** file to create a list of potential default gateways. The **/etc/dgateways** file consists of a series of lines, each in the following format:

gateway metric

where:

gateway is the name or address of a gateway to be used as a potential default gateway.

metric is a metric indicating the preference weighting of the gateway. 1 is the value to use for highest preference, 15 for the least. If no value is specified, *metric* defaults to the value 1.

There can be a maximum of 128 valid entries in the **/etc/dgateways** file - additional ones are ignored, but cause an error message. Duplicate gateway names or addresses are not allowed - only the first one encountered in the file is added to the table, and duplicates produce error messages.

After the list of gateways is created, **routed** selects the one with the lowest metric value to be used as the preferred default route. If there are multiple gateways available with the same metric value, it uses the one named first in the **/etc/dgateways** file.

routed then listens on udp port 520 for routing information packets. When a RIP *request* or *reply* packet is received, **routed** marks the gateway that sent the packet ALIVE. If the gateway has a better metric than the current default gateway, or has the same metric but is listed earlier in **/etc/dgateways,** the current default gateway is changed to the new gateway.

When a gateway is not heard from for 90 seconds, **routed** marks the gateway as DEAD, and if it was the current default gateway, selects a new default gateway if one is available.

In addition, when **routed** is running, it deletes dynamic routes, created by ICMP redirects, every 3 minutes.

USAGE

routed on The route daemon may be turned on at any time with the **routed on** command. This causes **routed** to read the **/etc/dgateways** file, and turn on RIP snooping, dynamic route timeouts, and default gateway selection. If **routed** is already running, this option causes it to reread the **/etc/dgateways** file, and reinitialize. By default, **routed** is invoked at boot time in **/etc/rc**.

routed off The route daemon may be turned off at any time with the **routed off** command. This stops all RIP snooping, default gateway selection, and dynamic route timeouts. The currently selected default gateway is not be deleted when **routed** is turned off.

routed status Displays the status of the default gateway list. This shows whether RIP snooping is active, the current list of default gateways, their metrics, the state of the gateways (ALIVE or DEAD), and the last time each gateway was heard from. The output looks like:

maytag> routed status

RIP snooping is on

Gateway	Metric	State	Time Last Heard
alantec1 Wed	1	ALIVE	Mar 9 03:38:41 GMT 1994
groucho Wed	1	ALIVE	Mar 9 03:38:41 GMT 1994
192.9.200.66	1	ALIVE	Wed Mar 9 03:38:41 GMT 1994
192.9.200.77	1	ALIVE	Wed Mar 9 03:38:41 GMT 1994
tphub1 Wed	2	ALIVE	Mar 9 03:38:41 GMT 1994
192.9.200.32	2	ALIVE	Wed Mar 9 03:38:41 GMT 1994
192.9.200.252	3	ALIVE	Wed Mar 9 03:38:41 GMT 1994
192.9.200.251	4	ALIVE	Wed Mar 9 03:38:41 GMT 1994
192.9.200.250	5	ALIVE	Wed Mar 9 03:38:41 GMT 1994

119 free gateway entries, 9 used

OPTIONS

-**n** If this option precedes **status**, the command displays numeric values for gateway names.

FILES

/etc/rc for default initialization

/etc/dgateways for the list of default gateways.

SEE ALSO

netstat, route, setup, dgateways, rc

DIAGNOSTICS

routed: *unable to allocate free entry* - too many valid entries were found in the **/etc/dgateways** file. Only the first 128 are used.

routed: *duplicate gateway entry not allowed* - a duplicate gateway name or address was found in the **/etc/dgateways** file. Only the first one found is used.

routed: *unable to open socket* - a networking error has prevented **routed** from initializing properly.

NAME

savecore - save a core dump

SYNOPSIS

savecore

DESCRIPTION

savecore is meant to be called near the end of the initialization file **/etc/rc**. Its function is to save the core dump of the system (assuming one was made) and to write the panic string to **/etc/messages**. **savecore** saves the core dump in two files **/etc/crash/core.***n*, and **/etc/crash/core.***n*-**small,** where *n* is determined by the **/etc/crash/bounds** file.

The -**small** core file contains a subset of the memory image that Dell can use for initial troubleshooting. Dell will only need to look at the large core file if the problem cannot be determined by examining the small one.

Before **savecore** writes out a core image, it reads a number from the file **/etc/crash/minfree**. If the number of free kilobytes in the filesystem after saving the core would be less than the number obtained from **minfree**, the core dump is not saved. If **minfree** does not exist, **savecore** always writes out the core file (assuming that a core dump was taken).

FILES

/etc/crash/core.* saved core files

/etc/crash/core.*-small
saved small core files

/etc/crash/bounds suffix for next core file

/etc/crash/minfree free KB in FS to maintain after savecore

SEE ALSO

rc

NAME

setup - update filer configuration

SYNOPSIS

setup

DESCRIPTION

setup queries the user for the filer configuration parameters such as hostname, IP address, and timezone. It installs new versions of **/etc/rc**, **/etc/hosts**, **/etc/exports**, **/etc/resolv.conf**, **/etc/hosts.equiv**, and **/etc/dgateways** to reflect the new configuration. When **setup** completes, the configuration files have been updated, but their new contents do not take effect until the filer is rebooted (see **reboot**). The old contents of the configuration files are saved in **rc.bak**, **exports.bak**, **resolv.conf.bak**, **hosts.bak**, **hosts.equiv.bak**, and **dgateways.bak**.

One piece of information that **setup** requests is the name and IP address for *adminhost*. In **/etc/exports**, *adminhost* is granted root access to **/** so that it can access and modify the configuration files in **/etc**. All other NFS clients are granted access only to **/home**. If no *adminhost* is specified, then all clients are granted root access to **/**. This is not recommended for sites where security is a concern.

If an adminhost is specified, then an additional line is added to the **/etc/hosts** file to point the default mailhost to the adminhost.

If a default gateway is provided to setup, it will be used in **/etc/rc** to specify a default route (see **route**), and will also be used as the first entry in **/etc/dgateways**.

The *hostname* that is provided to **setup** is used to construct default names for all of the configured network interfaces. Ethernet interfaces are given names *hostname*-**0**, *hostname*-**1**, and so on.

FILES

/etc	directory of filer configuration and administration files
/etc/rc	system initialization command script
/etc/exports	directories exported by the server
/etc/hosts	host name data base
/etc/hosts.equiv	list of hosts and users with rsh permission
/etc/resolv.conf	list of DNS name servers
/etc/dgateways	list of preferred default gateways for routed
/etc/nsswitch.conf	list of preferred name services

SEE ALSO

ifconfig, reboot, dgateways, exports, hosts, hosts.equiv, resolv.conf, rc, autosupport

NAME

shelfchk - verify the communication of environmental information between disk shelves and the filer

SYNOPSIS

shelfchk

DESCRIPTION

The **shelfchk** command verifies that the disk shelves and the filer can exchange environmental information. If the environmental information is being exchanged, you can hotswap disks in the disk shelves.

The **shelfchk** command is interactive. It requires that you type in your responses after observing the LEDs on the disks. Therefore, enter this command from a console that is near the disk shelves.

The **shelfchk** command steps through all the disk host adapters that the filer discovered when it booted. For each host adapter, the **shelfchk** command tries to turn on the disk LEDs on the attached disk shelves. The command waits for confirmation that you have observed the LEDs. If you see that all the LEDs are on, respond "yes" when prompted. If one or more LEDs are off, you respond "no" to the prompt. In this case, a problem exists that might prevent hot swapping on the affected shelves. The **shelfchk** command terminates as soon as you respond "no" to the prompt. It does not continue to test the other disk shelves. A possible cause of disk shelf problems is that the cables for the shelves are not connected properly.

Enter the **shelfchk** command immediately after you install one or more disk shelves. This way, if there are any cabling problems, you can fix them as soon as possible. Also, this command enables you to quickly correlate the disk shelves with their corresponding host adapter. For example, if you intend to have all disk shelves connected to a particular host adapter to be installed in one rack, the **shelfchk** command enables you to see at a glance whether any disk shelves were installed inadvertently in a different rack.

EXAMPLE

In the following example, the **shelfchk** command tests the disk shelves of a filer with three host adapters (8a, 8b, and 7a) and finds no problems:

```
filer> shelfchk
Only shelves attached to ha 7a should have all LEDs ON.
Are these LEDs all ON now? y
Only shelves attached to ha 8a should have all LEDs ON.
Are these LEDs all ON now? y
Only shelves attached to ha 8b should have all LEDs ON.
Are these LEDs all ON now? y
filer> Fri Aug 22 21:35:39 GMT [rc]: Disk Configuration - No Errors Identified
```

In the following example, the **shelfchk** command finds an error:

```
filer> shelfchk
Only shelves attached to ha 9a should have all LEDs ON.
Are these LEDs all ON now? n
*** Your system may not be configured properly. Check cable connections.
filer> Mon Aug 25 11:44:34 GMT [rc]: Disk Configuration - Failure Identified
by Operator
```

NAME

snap - manage snapshots

SYNOPSIS

snap list [*vol_name*]

snap create | **delete** *vol_name name*

snap rename *vol_name from to*

snap sched [*vol_name* [*weeks* [*days* [*hours*[@*list*]]]]]

snap reserve [*vol_name* [*percent*]]

DESCRIPTION

The **snap** family of commands provides a means to create and manage snapshots in each volume.

A snapshot is a read-only copy of the entire file system as of the time the snapshot was created. The filer uses a copy-on-write technique to create snapshots very quickly without consuming any disk space. Only as blocks in the active file system are modified and written to new locations on disk does the snapshot begin to consume extra space.

Snapshots are exported to all CIFS or NFS clients. They can be accessed from each directory in the file system. From any directory, a user can access the set of snapshots from a hidden sub-directory that appears to a CIFS client as **~snapsht** and to an NFS client as **.snapshot**. These hidden sub-directories are special in that they can be accessed from every directory, but they only show up in directory listings at an NFS mount point or at the root of CIFS share.

Each volume on the filer can have up to 20 snapshots at one time. Because of the copy-on-write technique used to update disk blocks, deleting a snapshot will generally not free as much space as its size would seem to indicate. Blocks in the snapshot may be referenced by other snapshots, or by the active file system, and thus may be unavailable for reuse even after the snapshot is deleted.

The **snap** commands are persistent across reboots. Do not include **snap** commands in the **/etc/rc**. If you include a **snap** command in the **/etc/rc** file, the same **snap** command you enter through the command line interface does not persist across a reboot and is overridden by the one in the **/etc/rc** file.

Automatic snapshots

Automatic snapshots can be scheduled to occur weekly, daily, or hourly. Weekly snapshots are named **weekly.***N*, where *N* is "0" for the most recent snapshot, "1" for the next most recent, and so on. Daily snapshots are named **daily.***N* and hourly snapshots **hourly.***N*. Whenever a new snapshot of a particular type is created and the number of existing snapshots of that type exceeds the limit specified by the **sched** option described below, then the oldest snapshot is deleted and the existing ones are renamed. If, for example, you specified that a maximum of 8 hourly snapshots were to be saved

using the **sched** command, then on the hour, **hourly.7** would be deleted, **hourly.0** would be renamed to **hourly.1**, and so on.

USAGE

snap list [*vol_name*]

displays a single line of information for each snapshot. Along with the snapshot's name, it shows when the snapshot was created and the size of the snapshot. If you include the *vol_name* argument, **list** displays snapshot information only for the specified volume. With no arguments, it displays snapshot information for all volumes in the system. The following is an example of the **snap list** output on a filer with two volumes named engineering and marketing.

Volume engineering

```
 %/used       %/total  date          name
--------  ---------  ------------  -------
 0% ( 0%)    0% ( 0%)  Nov 14 08:00  hourly.0
50% (50%)    0% ( 0%)  Nov 14 00:00  nightly.0
67% (50%)    0% ( 0%)  Nov 13 20:00  hourly.1
75% (50%)    0% ( 0%)  Nov 13 16:00  hourly.2
80% (50%)    0% ( 0%)  Nov 13 12:00  hourly.3
83% (50%)    1% ( 0%)  Nov 13 08:00  hourly.4
86% (50%)    1% ( 0%)  Nov 13 00:00  nightly.1
87% (50%)    1% ( 0%)  Nov 12 20:00  hourly.5
```

Volume marketing

```
 %/used       %/total  date          name
--------  ---------  ------------  -------
 0% ( 0%)    0% ( 0%)  Nov 14 08:00  hourly.0
17% (16%)    0% ( 0%)  Nov 14 00:00  nightly.0
28% (16%)    0% ( 0%)  Nov 13 20:00  hourly.1
37% (16%)    0% ( 0%)  Nov 13 16:00  hourly.2
44% (16%)    0% ( 0%)  Nov 13 12:00  hourly.3
49% (16%)    1% ( 0%)  Nov 13 08:00  hourly.4
54% (16%)    1% ( 0%)  Nov 13 00:00  nightly.1
58% (16%)    1% ( 0%)  Nov 12 20:00  hourly.5
```

snap create *vol_name name*

creates a snapshot of volume *vol_name* with the specified name.

snap delete *vol_name name*

deletes the existing snapshot belonging to volume *vol_name* that has the specified name.

snap rename *vol_name oldname newname*

gives an existing snapshot a new name. You can use the **snap rename** command to move a snapshot out of the way so that it won't be deleted automatically.

NAME

snmp - set and query SNMP agent variables

SYNOPSIS

snmp

snmp authtrap [**0** | **1**]

snmp community [**add** | **delete ro** | **rw**]

snmp contact [*contact*]

snmp init [**1**]

snmp location [*location*]

snmp traphost [**add** | **delete** *hostname* | *ipaddress*]

snmp traps [on | **off** | **reset** | **delete]**

snmp traps *trapname[.parameter value* | **on** | **off** | **reset delete]**

DESCRIPTION

The **snmp** command is used to set and query configuration variables for the SNMP agent daemon (see **snmpd**). If no options are specified, **snmp** lists the current values of all variables.

OPTIONS

In all the following options, specifying the option name alone prints the current value of that option variable. If the option name is followed by one or more variables then the appropriate action to set or delete that variable will be taken. Any variable with an inclusive space or tab must be enclosed in single quotes "' '".

It is recommended that all **snmp** commands be added to the end of the **/etc/rc** file. The last **snmp** command in the **/etc/rc** file should be:

snmp init 1

This will initialize the SNMP daemon with the values set using the **snmp** command and it will send out a **coldStart** trap as described below.

authtrap [0 | **1]**
Enable or disable SNMP agent authentication failure traps. To enable authentication traps specify **1**. To disable authentication traps specify **0**. Traps are sent to all hosts specified with the **traphost** option.

community [add|delete ro|rw *community* **]**

Add or delete communities with the specified access control type. Specify **ro** for a read-only community and **rw** for a read-write community. For example, to add the read-only community **private** use the following command:

snmp community add ro private

Currently the SNMP SetRequest PDU is not supported, so all read-write communities will default to read-only. The default community for the filer SNMP agent is **public** and its access mode is **ro**. Up to a maximum of 8 communities are supported.

contact [*contact* **]**

Used to set the contact name returned by the SNMP agent as the System.sysContact.0 MIB-II variable.

init [1] With an option of **1** this initializes the snmp daemon with values previously set by the snmp command. It also sends a **coldStart** trap to any hosts previously specified by the **traphost** option. The command:

snmp init 1

should be the last **snmp** command in the filer's **/etc/rc** file.

On a query **init** will return the value 0 if the SNMP daemon has not yet been initialized. Otherwise it will return the value 1.

location [*location*]

Used to set the location name returned by the SNMP agent as the System.sysLocation.0 MIB-II variable.

traphost [**add** | **delete** *hostname* | *ipaddress*]

To add or delete SNMP managers who will be the recipient of the filer's Trap PDU's. Specify the word **add** or **delete** as appropriate followed by the host name or address. If a host name is specified, it must exist in the **/etc/hosts** file. For example, to add the host **alpha** use the following command:

snmp traphost add alpha

No traps will be sent unless at least one trap host is specified. Up to a maximum of 8 trap hosts are supported.

On a query the **traphost** option will return a list of registered trap hosts followed by their IP addresses. If a host name cannot be found in **/etc/hosts** for a previously registered IP address, its name will default to a string representation of its IP address.

traps on | off | reset | delete

turns all traps on or off, or resets or deletes them

traps *trapname[.parameter value* | **on** | **off** | **reset** | **delete]**

affects a specified trap. It assigns a parameter and value, or turns it on or off, or resets or deletes it.

You can trap on any MIB variable, but to do so you need a trap data parser function at the traphost application or a management script to interpret the TRAP name, OID value fields, and so on.

EXAMPLES

A typical set of snmp commands in the **/etc/rc** file will look like the following:

```
snmp contact 'Network Manager'
snmp location 'Bldg 2. Lab 3a'
snmp community add ro private
snmp traphost add snmp-mgr1
snmp traphost add snmp-mgr2
snmp init 1
```

FILES

/etc/rc startup command script where snmp commands must be added

/etc/hosts hosts name database

SEE ALSO

autosupport, snmpd, rc

NAME

sysconfig - display filer configuration information

SYNOPSIS

sysconfig [**-d** | **-m** | **-r** | **-t** | [**-v**] [*slot*]]

DESCRIPTION

sysconfig displays the configuration information about the filer. Without any arguments, the output includes the Data ONTAP™ 5.3 version number and a separate line for each I/O device on the filer. If the *slot* argument is specified, **sysconfig** displays detail information for the specified physical slot; slot 0 is the system board, and slot *n* is the *n*th expansion slot on the filer.

OPTIONS

-d Displays vital product information for each disk.

-m Displays tape library information. To use this option, the autoload setting of the tape library must be off when the filer boots.

-r Displays RAID configuration information.

-t Displays device and configuration information for each tape drive.

-v Displays detailed information about each I/O device. For SCSI or Fibre Channel host adapters, the additional information includes a separate line describing each attached disk.

SEE ALSO

version, mt

NAME

sysstat - report filer performance statistics

SYNOPSIS

sysstat [*interval*]

DESCRIPTION

sysstat reports filer performance statistics such as the current CPU utilization, the amount of network I/O, the amount of disk I/O, and the amount of tape I/O. By default, **sysstat** prints a new line of statistics every 15 seconds. The *interval* argument overrides the default causing **sysstat** to report once every *interval* seconds. Use control-C to stop **sysstat**.

EXAMPLE

This is an example of **sysstat** running on a lightly loaded NFS-only filer:

```
filer> sysstat 1
CPU  NFS  CIFS  HTTP  Net kB/s      Disk kB/s      Tape kB/s  Cache
                 in    out   read  write  read write  age
 5%   82     0     0    15    17    16     0    0     0     8
 6%  105     0     0    24    98   100     0    0     0     8
 5%   54     0     0    32    11     0     0    0     0     8
21%   50     0     0    25    42   120   592    0     0     8
16%   27     0     0    10    10   144  1008    0     0     8
17%   90     0     0    64    11    16   104    0   552     8
15%   96     0     0    65    12     0     0    0   460     8
 5%   60     0     0    30    28    24     0    0     0     8
 1%   60     0     0    32    30    28     0    0     0     8
 4%   57     0     0    46    45    40     0    0     0     8
 5%   66     0     0    23    16     8     0    0     0     8
^C
filer>
```

From left to right, the columns indicate:

CPU the percentage CPU utilization during the previous *interval* seconds;

NFS	the number of NFS operations per second during that time;
CIFS	the number of CIFS operations per second during that time;
HTTP	the number of HTTP operations per second during that time;
Net kB/s	the number of kilobytes per second of network traffic into and out of the server;
Disk kB/s	the kilobytes per second of disk traffic being read and written;
Tape kB/s	the number of kilobytes per second of tape traffic being read and written;
Cache age	the age in minutes of the oldest read-only blocks in the buffer cache. Data in this column indicates how fast read operations are cycling through system memory; when the filer is reading very large files (larger than the machine's memory size), buffer cache age will be very low.

SEE ALSO

netstat, nfsstat

NAME

timezone - set the local timezone

SYNOPSIS

timezone [*name*]

DESCRIPTION

timezone sets the system timezone and saves the setting for use on subsequent boots. The argument *name* specifies the timezone to use. See the system documentation for a complete list of time zone names. If no argument is supplied, the current time zone name is printed.

Each timezone is described by a file that is kept in the **/etc/zoneinfo** directory on the filer. The *name* argument is actually the name of the file under **/etc/zoneinfo** that describes the timezone to use. For instance, the *name* "America/Los_Angeles" refers to the timezone file **/etc/zoneinfo/America/Los_Angeles**. These files are in standard "Arthur Olson" timezone file format, as used on many flavors of UNIX (SunOS 4.x and later, 4.4BSD, System V Release 4 and later, and others).

GMT+13 is to allow DST for timezone GMT+12.

FILES

/etc/zoneinfo directory of time zone information files

SEE ALSO

zoneinfo

NAME

uptime - show how long system has been up

SYNOPSIS

uptime

DESCRIPTION

uptime 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.

The filer runs **uptime** automatically once an hour and automatically logs its output to **/etc/messages**.

EXAMPLE

```
filer> uptime
  8:54am up 2 days 22:23, 3122520 NFS ops
```

SEE ALSO

netstat, nfsstat, sysstat, messages

NAME

useradmin - add, delete or list administrative users

SYNOPSIS

useradmin useradd *[-c comments] login_name*

useradmin userdel *login_name*

useradmin userlist *[user_name_list]*

DESCRIPTION

The **useradmin** command can be used to add, delete, or list administrative users.

The **useradd** option is used to add administrative users. The user name can be up to **32 characters** long. The user name can contain any printable **ASCII** characters except the following characters:

space | * + , / : ; < = > ? [\ ]

All users added through the **useradd** option have the same privilege level as root.

Optionally, any comment about the user being added could be provided. Comments about the user should be no longer than 128 characters and should not contain the character ':' (colon).

When users are added they are prompted for the password twice. The password is case sensitive and it has the following restrictions:

- it should be at least 6 characters long
- it should contain at least two alphabets
- it should contain at least one digit or special character

The **userdel** option can be used to delete any non-root administrative user.

The **userlist** option lists all non-root users if no user name is provided. The **userlist** option can also be invoked with a list of users to list information about only those users.

SEE ALSO

passwd

NAME

version - display Data ONTAP 5.3 version

SYNOPSIS

version

DESCRIPTION

version displays the version of Data ONTAP 5.3 running on the server, and the date when the version was created.

SEE ALSO

download, sysconfig

NAME

vif - create and destroy virtual interfaces

SYNOPSIS

vif create [*vif_name*] *interface_name* ...

vif destroy *vif_name*

vif stat *vif_name* [*interval*]

DESCRIPTION

The **vif** command creates and eliminates virtual interfaces. It also displays statistics for a specified virtual interface.

The **vif create** command creates a virtual interface. The name of the virtual interface to be created is specified as vif*n*, where *n* is a number. Make sure that the specified virtual interface name is not already in use. Use the **ifconfig** command to check and see what virtual interface names are being used.

You can specify up to four Ethernet interfaces in the command. The interfaces do not have to be on the same network card. However, some Ethernet switches or routers require that all Ethernet interfaces forming a virtual interface be either half-duplex or full-duplex. Check the documentation that comes with your Ethernet switch or router to see whether you need to configure the filer Ethernet interfaces to be half-duplex or full-duplex.

The **vif destroy** command eliminates an existing virtual interface. You must configure the virtual interface down using the **ifconfig** command before entering the **vif destroy** command.

The **vif stat** command displays the number of packets received and transmitted on each Ethernet interface that makes up the virtual interface. You can specify the time interval, in seconds, at which the statistics are displayed. By default, the statistics are displayed at one-second intervals.

EXAMPLES

The following example creates a virtual interface:

filer> **vif vif1 create e0 e7a e6b e8**

The following example eliminates virtual interface 1:

filer> **vif destroy vif1**

The following example displays statistics about virtual interface 1:

filer> **vif stat vif1**

Virtual interface (trunk) vif1

e5d		e5c		e5b		e5a	
In	Out	In	Out	In	Out	In	Out
8637076	47801540	158	159	7023083	38300325	8477195	47223431
1617	9588	0	0	634	3708	919	5400
1009	5928	0	0	925	5407	1246	7380
1269	7506	0	0	862	5040	1302	7710
1293	7632	0	0	761	4416	964	5676
920	5388	0	0	721	4188	981	5784
1098	6462	0	0	988	5772	1003	5898
2212	13176	0	0	769	4500	1216	7185
1315	7776	0	0	743	4320	530	3108

SEE ALSO

ifconfig

NAME

vol - commands for managing volumes, displaying volume status, and copying volumes

SYNOPSIS

vol *command argument ...*

DESCRIPTION

The **vol** commands manage a volume, apply options to a volume, or display the status of one or more volumes. Also, some **vol** commands copy volumes on the same filer or between two filers. You can use the volume copy feature only if you have purchased the volcopy license and entered the license code for a filer that is involved in a **vol copy** command.

USAGE

vol create *volname* [**-r** *raidsize*]
ndisks[@*size*] | [**-l** *language_code*] **-d** *diskname*... creates a new volume with the name *volname*. The volume name can contain letters, numbers, and the underscore character(_), but the first character must be a letter or underscore. You can create up to 23 volumes on each filer.

The **-r** *raidsize* argument specifies the maximum number of disks in each RAID group in the volume. The maximum value of *raidsize* is 28. The default value is 14. *ndisks* is the number of disks in the volume, including the parity disks. The disks in this newly created volume come from the pool of spare disks. The smallest disks in this pool join the volume first, unless you specify the @*size* argument. *size* is the disk size in GB, and disks within 1 GB of the specified size are used in this volume.

If you use the **-d** *diskname* argument, the filer creates the volume with the specified spare disks. You can specify a space-separated list of disk names.

If you use the **-l** *language_code* argument, the filer creates the volume with the language specified by the language code. The default is the language of the root volume of the filer. Language codes are:

C	(POSIX)
da	(Danish)
de	(German)
en	(English)
en_US	(English (US))
es	(Spanish)
fi	(Finnish)
fr	(French)
he	(Hebrew)
it	(Italian)
ja	(Japanese euc-j)
ja_JP.PCK	(Japanese PCK(sjis))
no	(Norwegian)

nl	(Dutch)
pt	(Portugese)
sv	(Swedish)

To use UTF-8 as the NFS character set append '.UTF-8'

vol add *volname ndisks*[*@size*] | **-d** *diskname*...
adds disks to the volume with the name *volname*. Specify the disks in the same way as for the **vol create** command.

When adding disks to a volume, the filer fills up one RAID group with disks before starting another RAID group. Suppose a volume currently has one RAID group of 12 disks and its RAID group size is 14. If you add 5 disks to this volume, it will have one RAID group with 14 disks and another RAID group with 3 disks. The filer does not evenly distribute disks among RAID groups.

vol destroy *volname*
destroys the volume with the name *volname*. The disks originally in the volume become spare disks. Only offline volumes can be destroyed.

vol lang [*volname* [*language_code*]
displays or changes character mapping on *volname*.

vol lang
by itself displays a list of supported languages.

vol lang *volname*
displays the language of the specified volume.

vol lang *volname language_code*
sets the language of volume *volname* to the language specified by *language_code*.

vol rename *volname newname*
renames the volume named *volname* to the name *newname*. If the volume named *volname* is referenced in the */etc/exports* file, remember to make the name change in */etc/exports* also so that the affected file system can be exported by the filer after the filer reboots. The **vol rename** command does not automatically update the */etc/exports* file.

vol online *volname*
brings the volume named *volname* online. This command takes effect immediately. The volume specified in this command must be currently offline or foreign. If the volume is foreign, it will be made native before being brought online. A "foreign" volume is a volume that consists of disks moved from another filer and that has never been brought online on the current filer. Volumes that are not foreign are considered "native." You can also use this command to cancel a **vol offline** command.

vol offline *volname*
takes the volume named *volname* offline. This command takes effect when

the filer is rebooted. If you change your mind after entering this command, you can enter **vol online** *volname* before the reboot.

vol options *volname optname optval*
sets the option named *optname* of the volume named *vol_name* to the value *optval*. The command remains effective after the filer is rebooted, so there is no need to add **vol options** commands to the */etc/rc* file. Some options have values that are numbers. Some options have values that may be **on** (which can also be expressed as **yes**, **true**, or **1**) or **off** (which can also be expressed as **no**, **false**, or **0**). You can use a mixture of uppercase and lowercase characters when typing the value of an option. The **root** option is special in that it does not have a value. To set the **root** option, use this syntax:

vol options *volname* **root**

The following describes the options and their possible values:

root The volume named *volname* will become the root volume for the filer on the next reboot. This option can be used on one volume only at any given time. The existing root volume will become a nonroot volume after the reboot. The only way to remove the root status of a volume is to set the **root** option on another volume.

raidsize *number*
The value of this option is the maximum size of a RAID group within the volume. Changing the value of this option will not cause existing RAID groups to grow or shrink; it will only affect whether more disks will be added to the last existing RAID group and how large new RAID groups will be.

minra on | **off**
If this option is **on**, the filer performs minimal read-ahead on the volume. By default, this option is **off**, causing the filer to perform very aggressive read-ahead on the volume.

no_atime_update on | **off**
If this option is **on**, it prevents the update of the access time on an inode when a file is read. This option is useful for volumes with extremely high read traffic, since it prevents writes to the inode file for the volume rom contending with reads from other files. It should be used carefully. That is, use this option when you know in advance that the correct access time for inodes will not be needed for files on that volume.

nosnap on | **off**
If this option is **on**, it disables automatic snapshots on the volume.

nosnapdir on | **off**
If this option is **on**, it disables the visible **.snapshot** directory that is normally present at client mount points, and turns off access to all other **.snapshot** directories in the volume.

nvfail on | **off**
If this option is **on**, the filer performs additional status checking at boot time to verify that the NVRAM is in a valid state. This option is useful when

storing database files. If the filer finds any problems, database instances hang or shut down, and the filer sends error messages to the console to alert you to check the state of the database.

snapmirrored off
If SnapMirror is enabled, the filer automatically sets this option to on. Set this option to off if you no longer want to use SnapMirror to update the mirror. After you set this option to off, the mirror becomes a regular writable volume. You can set this option only to off; only the filer can change the value of this option from off to on.

vol status [-**r** | -**v** | **-d** | **-l**] [*volname*] displays the status of one or all volumes. If *volname* is used, the status of the specified volume is printed; otherwise the status of all volumes in the filer are printed. By default, it prints a one-line synopsis of the volume, which includes the volume name, whether it is online or offline, other states (for example, partial, degraded, and so on) and per-volume options.

The -**v** flag displays information about each RAID group within the volume.

The -**r** flag displays a listing of the RAID information for that volume.

The -**d** flag displays information about the disks in the specified volume. The types of disk information are the same as those from the **sysconfig -d** command. The -**l** flag displays, for each volume on a filer, the name of the volume, the language code, and language being used by the volume.

vol copy start [**-S** | **-s**] *source destination* copies all data, including the snapshots, from one volume to another. If you use the **-S** flag, the command copies all snapshots in the source volume to the destination volume. To specify a particular snapshot to copy, use the **-s** flag followed by the name of the snapshot. If you use neither the **-S** nor **-s** flag in the command, the filer creates a snapshot at the time when the **vol copy start** command is executed and copies only that snapshot to the destination volume.

The source volume and destination volume can be on the same filer or different filers. If the source or destination volume is on a filer other than the one on which you enter the **vol copy start** command, specify the volume name in the *filer_name*:*volume_name* format.

The filers involved in a volume copy must meet the following requirements for the **vol copy start** command to be completed successfully:

The source volume must be on-line and the destination volume must be off-line.

If data is copied between two filers, each filer must be defined as a trusted host of the other filer. That is, the filer's name must be in the /etc/hosts.equiv file of the other filer.

If data is copied on the same filer, localhost must be included in the filer's /etc/hosts.equiv file. Also, the loopback address must be in

the filer's /etc/hosts file. Otherwise, the filer cannot send packets to itself through the loopback address when trying to copy data.

The usable disk space of the destination volume must be greater than or equal to the usable disk space of the source volume. Use the **df** *pathname* command to see the amount of usable disk space of a particular volume.

Each **vol copy start** command generates two volume copy operations: one for reading data from the source volume and one for writing data to the destination volume. Each filer supports up to four simultaneous volume copy operations.

vol copy abort [*operation_number*]
terminates a volume copy operation. The *operation_number* parameter in the **vol copy abort** command specifies which operation to terminate. If you don't specify an operation number, all volume copy operations are terminated.

vol copy status [*operation_number*]
displays the progress of one or all volume copy operations. The operations are numbered from 0 through 3.

vol copy throttle [*operation_number*] *value*
controls the performance of the volume copy operation. The value ranges from 10 (full speed) to 1 (one-tenth of full speed). You can apply the performance value to an operation specified by the *operation_number* parameter. If you do not specify an operation number in the **vol copy throttle** command, the command applies to all volume copy operations. Use this command to limit the speed of the volume copy operation if you suspect that the volume copy operation is causing performance problems on your filer.

The **vol copy throttle** command enables you to set the volume copy speed for a volume copy operation that is in progress. To set the default volume copy speed to be used by future volume copy operations, use the **options** command to set the **vol.copy.throttle** option.

vol snaprestore *volname* [-s *snapshot*] [*volname* [-s *snapshot*]...] reverts a volume to a specified snapshot. If you do not specify a snapshot, the filer prompts you for the snapshot. You can use one command to revert multiple volumes. The volume to be reverted must be on-line and must not be a mirror.

After the reversion, the volume is in the same state as it was when the snapshot was taken.

vol snapmirror on | off
enables and disables SnapMirror. If SnapMirror is enabled, you can replicate one volume to another according to the specifications in the **/etc/snapmirror.conf** file.

vol snapmirror status
displays whether SnapMirror is enabled. If so, the command displays whether the filer is copying data between the source volume and the mirror, and the percentage of data that has been copied.

EXAMPLES

vol create vol1 -r 10 20
creates a volume named **vol1** with 20 disks. The RAID groups in this volume can contain up to 10 disks, so this volume has two RAID groups. The filer adds the current spare disks to the new volume, starting with the smallest disk.

vol create vol1 20@9
creates a volume named **vol1** with 20 9-GB disks. Because no RAID group size is specified, the default size (14 disks) is used. The newly created volume contains one RAID group with 14 disks and another group with six disks.

vol create vol1 -d 8a.1 8a.2 8a.3
creates a volume named **vol1** with the specified disks.

vol create vol1 10
vol options vol1 raidsize 5
The first command creates a volume named **vol1** with 10 disks, which belong to one RAID group. The second command specifies that if any disks are subsequently added to this volume, they will not cause any current RAID group to have more than five disks. Each existing RAID group will continue to have 10 disks and no more disks will be added to that RAID group. When new RAID groups are created, they will have a maximum size of five disks.

vol options vol1 root
The volume named **vol1** becomes the root volume after the next filer reboot.

vol options vol1 nosnapdir on
In the volume named **vol1**, the snapshot directory is invisible at the client mount point or at the root of a share. Also, for UNIX clients, the .snapshot directories that are normally accessible in all the directories become inaccessible.

vol status -r vol1
displays the RAID information about the volume named **vol1**:

RAID group 0

RAID Disk	HA.DISK_ID	Used (MB/blks)	Phys (MB/blks)
parity	0.3	4000/8192000	4095/8386728
data	0.2	4000/8192000	4095/8386728

vol copy start -s nightly.1 vol0 filer1:vol0 copies the nightly snapshot named nightly.1 on the vol0 volume on the local filer to the vol0 volume on a remote filer named filer1.

vol copy status
displays the status of all the volume copy operations.

vol copy abort 1
terminates volume copy operation 1.

vol copy throttle 1 10
changes volume copy operation 1 to one-tenth of its full speed.

SEE ALSO

sysconfig

NAME

ypwhich - display the NIS server if NIS is enabled

SYNOPSIS

ypwhich

DESCRIPTION

ypwhich prints the name of the current NIS server if NIS is enabled. If there is no entry for the server itself in the hosts database, then it prints the IP address of the server.

The NIS server is dynamically chosen by the filer.

File Formats

This section contains file formats.

NAME

tape - information on filer tape interface

DESCRIPTION

The filer supports up to four local tape drives (tape drives connected directly to the system). The tape drive interface follows a UNIX-like device name allowing use of a **rewind**, **norewind** or **unload/reload** device. The format of a filer tape device name is *c***rst***ud* where:

c use **n** to specify the **norewind** device, use **u** to specify the **unload/reload** device, or no flag to specify the **rewind** device. The **norewind** device will not rewind when the tape device is closed. The **unload/reload** device is used with sequential tape loaders and will unload the current tape volume and attempt to load the next tape volume (note that the server will wait up to one minute for the next volume to become ready before aborting the reload of the next volume). The rewind device will rewind the tape volume to beginning-of-tape on close.

rst the **rst** portion of the device name is always present and specifies that you are requesting a SCSI tape device.

u the logical unit number of the tape drive to use.

d the density (or format) to use for tape write operations.

The density specifications for an Exabyte 8505 8mm drive are:

l Exabyte 8200 format, no compression

m Exabyte 8200 format with compression

h Exabyte 8500 format, no compression

a Exabyte 8500 format with compression

EXAMPLES

The **sysconfig -t** command will display the supported tape drives on your system and the device names associated with each tape device along with the device's density, or format. The following is an example of the output from a sysconfig command on a filer with one tape device attached:

```
filer> sysconfig -t

Tape drive (0.6)  Exabyte 8505 8mm
rst0l  -    rewind device,          format is: EXB-8200   2.5GB
nrst0l -    no rewind device,       format is: EXB-8200   2.5GB
urst0l -    unload/reload device,   format is: EXB-8200   2.5GB
rst0m  -    rewind device,          format is: EXB-8200C  (w/compression)
```

nrst0m -	no rewind device,	format is: EXB-8200C (w/compression)
urst0m -	unload/reload device,	format is: EXB-8200C (w/compression)
rst0h -	rewind device,	format is: EXB-8500 5.0GB
nrst0h -	no rewind device,	format is: EXB-8500 5.0GB
urst0h -	unload/reload device,	format is: EXB-8500 5.0GB
rst0a -	rewind device,	format is: EXB-8500C (w/compression)
nrst0a -	no rewind device,	format is: EXB-8500C (w/compression)
urst0a -	unload/reload device,	format is: EXB-8500C (w/compression)

SEE ALSO

dump, **mt, sysconfig**

Headers, Tasks, and Macros

This section contains headers, tasks, and macros.

NAME

boot - directory of Data ONTAP 5.3 executables

SYNOPSIS

/etc/boot

DESCRIPTION

The **boot** directory contains copies of the executable files required to boot the filer. The **download** command (see **download**) copies these files from **/etc/boot** into the filer's boot block, from which the system boots.

FILES

/etc/boot	directory of Data ONTAP 5.3 executables
/etc/boot/netapp-alpha	symbolic link to current version of Data ONTAP 5.3 for filers with Alpha processors
/etc/boot/fc-hard-alpha	boot FCode for filers with Alpha processors
/etc/boot/1-alpha	second stage boot code for filers with Alpha processors

SEE ALSO

download

NAME

crash - directory of system core files

SYNOPSIS

/etc/crash

DESCRIPTION

If a filer crashes, it creates a core file in the **crash** directory. The core files are very useful for finding and fixing bugs in Data ONTAP 5.3, so notify Dell technical support of any core files on your filer.

See **savecore** for more details about how core files are saved.

FILES

/etc/crash/core.*	saved core files
/etc/crash/core.*-small	compact core file.
/etc/crash/bounds	suffix for next core file
/etc/crash/minfree	free KB in FS to maintain after savecore

SEE ALSO

savecore

NAME

dgateways - default gateways list

SYNOPSIS

/etc/dgateways

DESCRIPTION

The **/etc/dgateways** file is used by the **routed** command to construct a set of potential default gateways. The file is comprised of a series of lines, each in the following format:

gateway metric

gateway is the name or address of a gateway to be used as a potential default gateway.

metric is a metric indicating the preference weighting of the gateway. 1 is the value to use for highest preference, 15 for the least. If no value is specified, *metric* will default to the value 1.

There can be a maximum of 128 valid entries in the **/etc/dgateways** file - additional ones will be ignored, with an error message being displayed. Duplicate gateway names or addresses are not allowed - only the first one encountered in the file will be added by **routed** to the default gateway table, and the additional ones will produce error messages.

EXAMPLE

Here are typical lines from the **/etc/dgateways** file:

```
main_router    1
backup_router  2
```

SEE ALSO

routed, setup

NAME

dumpdates - data base of file system dump times

SYNOPSIS

/etc/dumpdates

DESCRIPTION

The **dump** command (see **dump**) uses **/etc/dumpdates** to keep track of which subtrees have been dumped and when. Each line in **dumpdates** contains the subtree dumped, the dump level, and the creation date of the snapshot used by **dump**. There is only one entry per subtree at a given dump level. **dumpdates** may be edited to change any of the fields, if necessary.

EXAMPLE

This shows the dumpdate file for a system on which **/home** and **/export** are backed up using **dump**.

```
/home    0 Tue  Nov  2  10:56:27    1993
/export  0 Tue  Nov  2  13:51:17    1993
/export  1 Tue  Nov  5  18:31:17    1993
/home    1 Tue  Nov  5  18:45:27    1993
```

FILES

/etc/dumpdates

SEE ALSO

dump

NAME

exports - directories and files exported to NFS clients

SYNOPSIS

/etc/exports

DESCRIPTION

The **/etc/exports** file contains a list of directories and files that are exported by the filer. Changes to this file do not take effect until the filer executes the **exportfs** command or the filer is rebooted. When the filer is rebooted, it executes the **exportfs** -**a** command from the **/etc/rc** script to export all files and directories listed in the **/etc/exports** file.

Each export entry is a line in the following format:

pathname -*option*[,*option*] ...

The following list describes the fields in an export entry:

pathname path name of a file or directory to be exported.

option the export option specifying how a file or directory is exported. You can specify the option in one of the following formats:

access=*hostname*[:hostname]...

Give mount access to each host listed. Alternatively, you can specify a netgroup instead of a host in the list. The netgroup must be defined in the **/etc/netgroup** file. Whether the hosts can mount *pathname* with root access, read-and-write access, or read-only access depends on how you use the **root**, **rw**, and **ro** options, as described below.

anon=*uid*

If a request comes from user ID of 0 (root user ID on the client), use *uid* as the effective user ID unless the client host is included in the **root** option. The default value of *uid* is 65534. To disable root access, set *uid* to 65535. To grant root access to all clients, set *uid* to 0.

ro

Export the *pathname* read-only. If you do not specify this option, the *pathname* is exported read-write.

rw=*hostname*[:hostname]...

Export the *pathname* read-only to all hosts not specified in the list and read-write to the hosts in the list. Netgroup names are not allowed in the list.

root=*hostname*[:hostname]...

Give root access only to the specified hosts. By default, no hosts are granted root access. Netgroup names are not allowed in the list.

In an export entry, you can specify that a file or directory be exported to a subnet instead of individual hosts. The export entry for exporting to subnets can use the **ro**, **rw**, or **root** option; you cannot specify a subnet in the list for the **access** option.

Instead of specifying a host name or netgroup name in the entry, specify the subnet in one of the following formats:

dotted_IP/num_bits The *dotted_IP* field is either an IP address or a subnet number. The *num_bits* field specifies the size of the subnet by the number of leading bits of the netmask.

"[**network**] *subnet* [**netmask**] *netmask*"
The *subnet* field is the subnet number. The *netmask* field is the netmask.

In UNIX, it is illegal to export a directory that has an exported ancestor in the same file system. Data ONTAP 5.3 does not have this restriction. For example, you can export both the **/** directory and the **/home** directory. In determining permissions, the filer uses the longest matching prefix.

EXAMPLES

In the following example, all network clients can mount the **/home** directory but only the **adminhost** can mount the **/** directory:

```
/    -access=adminhost,root=adminhost
/home
```

The following examples show different ways of specifying an export entry that exports the **/home** directory to the 123.45.67.0 subnet with the 255.255.255.0 netmask:

```
/home    -rw=123.45.67.0/24
/home    -rw=123.45.67/24
/home    -rw="network 123.45.67.0 netmask 255.255.255.0"
/home    -rw="123.45.67.0 255.255.255.0"
```

FILES

/etc/exports	directories and files exported to NFS clients
/etc/hosts	host name database

SEE ALSO

exportfs, reboot, hosts, netgroup, rc

NAME

group - group file

SYNOPSIS

/etc/group

DESCRIPTION

The **/etc/group** database contains information for each group in the following form:

groupname:password:gid:user-list

The following list describes the required fields:

groupname	The name of the group.
password	The group's password, in an encrypted form. This field may be empty.
gid	An interger representing the group; each group is assigned a unique integer.
user-list	The user list is a comma-separated list of users allowed in the group.

EXAMPLE

Here is a sample group file:

project:asderghuloiyw:12:dan,dave
myproject::11:steve,jerry

SEE ALSO

nis, nsswitch.conf, quota, cifs_access, cifs_setup

NAME

hosts - host name data base

SYNOPSIS

/etc/hosts

DESCRIPTION

The **hosts** file contains information regarding the known hosts on the network. For each host a single line should be present with the following information:

Internet-address official-host-name aliases

Items are separated by any number of blanks and/or tab characters. A "#" indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines which search the file.

This file may be created from the official host data base maintained at the Network Information Control Center (NIC), though local changes may be required to bring it up to date regarding unofficial aliases and/or unknown hosts.

Network addresses are specified in the conventional "." (dot) notation. Host names may contain any alphanumeric character, but not field delimiters, newline, or comment characters.

FILES

/etc/hosts

SEE ALSO

hostname, dns, nis

NAME

hosts.equiv - list of hosts and users with rsh permission

SYNOPSIS

/etc/hosts.equiv

DESCRIPTION

The **hosts.equiv** file contains a list of hosts on which you can enter a filer command through the remote shell protocol (**rsh**).

Hosts specified in this file are considered the trusted hosts of the filer.

Each line in **hosts.equiv** has the following format:

hostname [*username*]

If the host on which you enter the filer command is a UNIX host, the user name is optional. If the host on which you enter the filer command is a PC, you must enter the user name for that PC in the **/etc/hosts.equiv** file.

If you do not specify a user name for a UNIX host, you must be root on that host to execute a filer command through **rsh**.

When you use an **rsh** application on your PC to issue a filer command, specify that you are entering the command as root.

If multiple users on the same host should have access to the filer through **rsh**, enter each user name on a separate line.

EXAMPLE

The following **hosts.equiv** file allows both **root** and **joe_smith** to enter filer commands through **rsh** on a UNIX host named **adminhost**:

adminhost
adminhost joe_smith

SEE ALSO

rshd

NAME

httpd.access - authentication controls for HTTP access

SYNOPSIS

/etc/httpd.access

DESCRIPTION

The HTTP daemon can apply authentication controls to individual users or groups on a per directory basis. The file **/etc/httpd.access** specifies the following items for each access-controlled tree:

the path to the tree

the authority required to authenticate access to the tree

the lists of users or groups to who are permitted access when authenticated

The syntax is the same as the access control syntax used by NCSA and Apache. However, the **httpd.access** file only supports a subset of directives supported by NCSA and Apache. You can copy an existing NCSA or Apache access to the filer without editing or reformatting.

SYNTAX

The supported directives are:

<Directory directory_name>
</Directory>
AuthName *Title phrase*
require user *user_id* [, *user_id,...*]
require group *group_id* [, *group_id,...*]

where *Title phrase* is a word or phrase that is passed to the authentication dialog as a title for the dialog that prompts the user for a password.

EXAMPLES

The following example restricts access to the file **/home/htdocs/private/bob** so that only user dole can access it, after supplying the required password. The authentication dialog is titled "My private stuff."

```
<Directory /home/htdocs/private/bob>
AuthName My private stuff
<Limit GET>
require user dole
</Limit>
</Directory>
```

The **<Limit GET>** and **</Limit>** directives are not supported, but are retained for format consistency with NCSA and Apache. The filer just ignores them.

The following example restricts access to the directory tree **/home/htdocs/private/storage** to the group "group1," which consists of the users whose IDs are user1, user2, user3, and user4. The authentication dialog is titled "Dell."

```
<Directory /home/htdocs/private/storage>
AuthName Dell
<Limit GET>
require group group1
</Limit GET>
</Directory>
```

In this example, "group1" is defined by the following entry in **/etc/httpd.group**:

group1: user1 user2 user3 user4

SEE ALSO

httpd.passwd, httpd.group

NAME

httpd.group - names of HTTP access groups and their members

SYNOPSIS

/etc/httpd.group

DESCRIPTION

The file declares the names of groups, and the user IDs of the members of each group, for use by the HTTP daemon in executing the access controls declared in **/etc/httpd.access**.

SYNTAX

group_id1:*user_id1* [*user_id2* ...]

SEE ALSO

httpd.access

NAME

httpd.hostprefixes - configuration of HTTP root directories for virtual hosts

SYNOPSIS

/etc/httpd.hostprefixes

DESCRIPTION

The **httpd.hostprefixes** file maps virtual hosts used in HTTP to corresponding root directories. The same configuration file is used for both IP virtual hosts (discriminated by the IP address used for connecting to the server) and HTTP virtual hosts (discriminated by the **Host:** header used in HTTP requests).

Each virtual host has a corresponding subdirectory within the directory specified by the option **httpd.rootdir**. This subdirectory is called the virtual host root directory. Clients connected to a virtual host can only access files within the virtual host root directory.

In the **httpd.hostprefixes** file, each line consists of a virtual host root directory followed by the names and IP addresses of a virtual host. If you specify an IP address, the virtual host root directory is associated with the given virtual host for IP-level virtual hosting. If you specify a name, the virtual host root directory is associated with the virtual host with that name, using HTTP-level virtual hosting. If the filer can resolve that name to an IP address, which is used for an IP-level host alias (see the **alias** option in **ifconfig**), the filer uses that IP address in the same way as it would if you specified the IP address in the **httpd.hostprefixes** file.

If the **/etc/httpd.hostprefixes** file is edited, it is read again by the HTTP server after the changes are saved.

EXAMPLE

This example maps requests sent to **www.customer1.com** to the **customer1** subdirectory of **httpd.rootdir** and requests directed at a host with IP address 207.68.156.58 to the subdirectory **customer2**.

```
/customer1 www.customer1.com
/customer2 207.68.156.58
```

If the command

```
filer>ifconfig vh alias www.customer1.com
```

had been issued before the configuration file was read, requests destined for the IP address of **www.customer1.com** would also be mapped to the **/customer1** subdirectory, regardless any the **Host:** header they included.

SEE ALSO

ifconfig, options

NAME

httpd.log - Log of HTTP

SYNOPSIS

/etc/log/httpd.log

DESCRIPTION

The HTTP server logs an entry for every file retrieved via HTTP. This log, written to **/etc/log/httpd.log**, is stored in the "Common Log Format," which is used by many World-Wide Web servers.

Each entry in **/etc/log/httpd.log** consists of one line with seven fields. The fields are, in order:

address The IP address of the HTTP client requesting the file.

rfc931 This field is always "-".

authuser This field is always "-".

date The time and date the request was is reported in the format "[Day/Mon/Year:HH:MM:SS]", which is logged in universal time (GMT) rather than the local time zone.

request A quoted string is recorded for the method (request type) and file involved in the request.

result The status code for the request.

bytes The size of the file in bytes.

Possible values for *result* codes include:

200 Success: the requested file was transmitted.

302 Redirected (see **/etc/httpd.translations**)

304 Not modified (client cache used)

400 Bad request.

403 Access to file prohibited.

404 File not found.

503 HTTP server disabled.

The size of the log file can be restricted by the option **httpd.log.max_file_size**.

SEE ALSO

options, httpd.translations

NAME

httpd.mimetypes - map of file suffixes to MIME Content-Type

SYNOPSIS

/etc/httpd.mimetypes

DESCRIPTION

For HTTP/1.0 and higher protocols, a MIME header is returned in the reply of every GET request. This header includes a "Content-Type" field, whose contents is determined by examining the suffix of the file being transmitted.

The **/etc/httpd.mimetypes** file contains the mapping of filename suffixes to MIME Content-Type. The format of each line is: suffix, Content-Type. Comments are introduced with a "#".

The filer is not shipped with the **/etc/httpd.mimetypes** file. Instead, the filer's system files include a sample file named **/etc/httpd.mimetypes.sample**. Before you start using HTTP, make a copy of **/etc/httpd.mimetypes.sample** and name the copy **/etc/httpd.mimetypes**.

If the file **/etc/httpd.mimetypes** is not installed, the HTTP server looks for the file **/etc/httpd.mimetypes.sample** as a fallback.

EXAMPLE

```
#map .ps files to PostScript
ps application/postscript
```

NAME

httpd.passwd - file of passwords required for HTTP access

SYNOPSIS

/etc/httpd.passwd

DESCRIPTION

The password file containing the encrypted form of the password that an HTTP client must supply to have access to a file in a controlled-access directory tree, as declared in **/etc/httpd.access**.

The password is encrypted in the regular UNIX style. User of NCSA or Apache can use their **htpasswd** program to generate the user_id:passwd pair.

The HTTP access control does not use the existing CIFS password database on the filer because in http basic authentication, in each request for protected pages, the value of *passwd* is sent over the network in clear text, and without encryption would compromise the user's password.

NOTE:

Encrypted password file can only be generated and imported from a UNIX client.

SYNTAX

user_id1:encrypted_passwd1
used_id2:encrypted_passwd2
...

SEE ALSO

httpd.access

NAME

httpd.translations - URL translations to be applied to incoming HTTP requests

SYNOPSIS

/etc/httpd.translations

DESCRIPTION

The HTTP daemon supports four URL translation rules to filter incoming HTTP requests. The HTTP daemon applies each rule in succession, stopping at the first successful **Redirect**, **Pass**, or **Fail** rule:

Map *template result*
Any request which matches *template* is replaced with the *result* string given.

Redirect *template result*
Any request which matches *template* is redirected to the *result* URL. Note that this must be a full URL, e.g., beginning with "http:".

Pass *template* [*result*]
Any request which matches *template* is granted access, and no further rule processing occurs. An optional *result* can be used in place of the matching URL.

Fail *template*
Any request which matches *template* is denied access. Rule processing stops after a matched **Fail**.

Both templates and results may contain wildcards (a star "*" character). The wildcard behaves like a shell wildcard in the *template* string, matching zero or more characters, *including* the slash ("/") character. In the *result* string, a wildcard causes text from the corresponding match in the *template* string to be inserted into the result.

EXAMPLE

This example redirects CGI queries to **cgi-host**, prevents accesses to **/usr/forbidden**, and maps requests to images to a local image directory:

```
#
# Example URL translations
#
Redirect /cgi-bin/* http://cgi-host/*
Fail /usr/forbidden/*
Map /image-bin/* /usr/local/http/images/*
```

NAME

messages - record of recent console messages

SYNOPSIS

/etc/messages

DESCRIPTION

The default behavior of the filer **syslogd** daemon (see **syslogd**) is to print all logging messages of priority **info** or higher to the console, and to the **messages** file.

Every Saturday at 24:00, **/etc/messages** is moved to **/etc/messages.0**, **/etc/messages.0** is moved to **/etc/messages.1**, and so on. Message files are saved for a total of six weeks.

FILES

/etc/messages	messages file for current week
/etc/messages.[0-5]	messages file for previous weeks

SEE ALSO

syslogd, syslog.conf

NAME

netgroup - network groups data base

SYNOPSIS

/etc/netgroup

DESCRIPTION

netgroup defines network wide groups used for access permission checking during remote mount request processing. Each line defines a group and has the format:

groupname member-list

Each element in member-list is either another group name or a triple of the form:

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

The *hostname* entry must be fully qualified if the specified host is not in the local domain.

The filer can also use the **netgroup** NIS map.

Since the filer uses netgroups only in **/etc/exports** (see **exports**), the *username* entry is ignored. The *domain_name* field refers to the domain in which the net-group entry is valid. It must either be empty or be the local domain, otherwise the netgroup entry is ignored. This allows a single **/etc/netgroup** file to be used for filers in multiple domains.

EXAMPLE

This is a typical **netgroup** file:

trusted_hosts (adminhost,,) (group1,,) (group2,,) (group3,,)

untrusted_hosts (group4,,) (group5,,) (group6,,)

all_hosts trusted_hosts untrusted_hosts

With this **netgroup** file it might make sense to modify **/etc/exports** to export **/** on the filer only to *trusted_hosts*, but to export **/home** to *all_hosts*.

FILES

/etc/netgroup

/etc/exports directories and files exported to NFS clients

/etc/hosts host name data base

SEE ALSO

exportfs, hosts, exports, nis

NAME

networks - network name data base

SYNOPSIS

/etc/networks

DESCRIPTION

The **networks** file contains information regarding the known networks which comprise the Internet. For each network a single line should be present with the following information:

official-network-name network-number aliases

Items are separated by any number of blanks and/or tab characters. A "#" indicates the beginning of a comment; characters up to the end of the line are not interpreted by routines which search the file. This file is normally created from the official network data base maintained at the Network Information Control Center (NIC), though local changes may be required to bring it up to date regarding unofficial aliases and/or unknown networks.

Network number may be specified in the conventional "." (dot) notation or as a 32 bit integer. Numbers may be specified in decimal (default), octal or hexadecimal. A number is interpreted as octal if it starts with the digit "0". A hexadecimal number must begin with "0x" or "0X." Network names may contain any printable character other than a field delimiter, newline, or comment character.

FILES

/etc/networks

NAME

nsswitch.conf - configuration file for name service switch

SYNOPSIS

/etc/nsswitch.conf

DESCRIPTION

The name service switch configuration file contains the preferred order in which name services will be contacted for name resolution by the filer. For each map, the name services to be used and the lookup order is specified in this file. Currently three name services are supported. They are local files in the /etc directory, NIS and DNS. The maps or "databases" that are supported are hosts, passwd, shadows, group and netgroups. Each line has the form:

map: order of name services

For example:

hosts: files nis dns
passwd: files nis

When trying to resolve a name, the services are contacted one by one, as per the order specified, until the name is successfully resolved. A name resolution failure occurs when no service can successfully resolve the name. When enumerating a map, enumeration happens over all the services specified for the map.

FILES

/etc/nsswitch.conf

SEE ALSO

setup

NAME

passwd - password file

SYNOPSIS

/etc/passwd

DESCRIPTION

The **passwd** file contains basic information about each user's account. It contains a one-line entry for each authorized user, of the form:

username:password:uid:gid:gcosfield:home_directory:login_shell

Required Fields:

username — The user's login name, not more than eight characters.

password — The user's password, in an encrypted form that is generated by the UNIX passwd function. However, if the encrypted password is stored in **/etc/shadow,** (see **shadow**), the password field of **/etc/passwd** is empty.

uid — A unique integer assigned by the UNIX administrator to represent the user's account; its value is usually between 0 and 32767.

gid — An integer representing the group to which the user has been assigned. Groups are created by the UNIX system administrator; each is assigned a unique integer whose value is generally between 0 and 32767.

gcos-field — The user's real name. The name may be of any length; it may include capital letters as well as lower case, and may include blanks. The name may be empty.

home_directory — The user's home directory. The home directory field may be empty.

login-shell — The default shell launched at login. This field may be empty.

EXAMPLE

Here is a sample **passwd** file when **the /etc/shadow** does not exist:

root:bDPu/ys5PBoYU:0:1:Operator:/:/bin/csh
dave:Qs5l6pBb2rJDA:1234:12:David:/u/dave:/bin/csh
dan:MNRWDsW/srMfE:2345:23:Dan::
jim:HNRyuuiuMFerx:::::

If the system keeps the passwords in the **/etc/shadow,** the file **/etc/passwd** would be exactly the same but the password field would be empty.

root::0:1:Operator:/:/bin/csh
dave::1234:12:David:/u/dave:/bin/csh

```
dan::2345:23:Dan::
jim::::::
```

SEE ALSO

shadow, options, nis, nsswitch.conf, quota, cifs_access, cifs_setup

NAME

quotas - quota description file

SYNOPSIS

/etc/quotas

DESCRIPTION

The /etc/quotas file describes disk quotas that go into effect when quotas are enabled. All quotas are established on a per-volume basis. If a volume name is not specified in an entry of the /etc/quotas file, the entry applies to the root volume.

The following sample /etc/quotas file describes different kinds of quotas:

Quota Target	type	disk	files
mhoward	user	500M	50K
lfine	user@/vol/home	500M	
tracker	user	-	-
stooges	group@/vol/vol0	750M	75K
/vol/vol0/export	tree	750M	75K
mhoward	user@/vol/vol0/export	50M	5K
stooges	group@/vol/vol0/export	100M	10K
*	user@/vol/home	100M	10K
*	group@/vol/vol0	500M	70K
*	user@/vol/vol0/export	20M	2K
*	group@/vol/vol0/export	200M	20K

The first non-comment line in the file restricts the user mhoward to 500 MB of disk space and 51,200 files in the root volume. The second line restricts the user lfine to 500 MB of disk space in the home volume, but places no restriction on the number of files he can have. You can leave the file limit blank to indicate that no limit is imposed but you cannot omit the value for disk space. The third line places no restriction on either disk usage or file usage by using a limit field of "-". This may be useful for tracking usage on a per-user or per-group basis without imposing any usage limits.

The next two lines restrict the stooges group and the /vol/vol0/export qtree to 750 MB and 76,800 files each in the root volume.

A user or group is specified by one of the following values:

a user or group name, which must appear in the password or group database (either in the /etc/passwd or /etc/group file on the filer, or in the password or group NIS map if NIS is enabled on the filer and is being used for the password or group database);

a numerical user or group ID;

the pathname of a file owned by that user or group.

The user or group identifier for a user or group quota can be followed by an **@/vol/***volume* string, which specifies the volume to which the quota applies. If the string is omitted, the quota applies to the root volume.

A quota of type **tree** can only be applied to a qtree, which is a directory in the root directory of a specified volume. A qtree is created with the **qtree create** or **quota qtree** command.

User and group quotas can be created inside a qtree, so that the user's or group's use of space or files within that qtree is restricted. This is done by specifying the type as **user@***tree* or **group@***tree* where *tree* is the name of the qtree. In the example above, we first limit overall usage in the qtree /vol/vol0/export and then we restrict the user mhoward to 50 MB and 5,120 files under the /vol/vol0/export tree. Similarly, the group stooges has been limited to 100 MB of disk space and 10,240 files under the /vol/vol0/export tree.

In any operation that creates files or writes to them, all applicable quotas must be satisfied. For example, the user mhoward can write to a file in the /vol/vol0/export tree if all of these requirements are met:

his total disk usage in the root volume does not exceed 500 MB

his total number of files in the root volume does not exceed 51,200

his usage within the /vol/vol0/export tree does not exceed 50 MB

his number of files within the /vol/vol0/export tree does not exceed 5,120

the space already in use in the /vol/vol0/export tree does not exceed 750 MB

the number of files in the /vol/vol0/export tree does not exceed 768,000

The asterisk (*) in the /etc/quotas file specifies a default user or a group quota depending on the type. Any user or group that is not specifically mentioned in the /etc/quotas file is subject to the limits of the default user or group. Default user or group quotas can be specified on a per qtree basis or a per volume basis.

Disk and file size limits in the third and fourth columns of the /etc/quotas file ends in "K", "M", or "G". "K" indicates kilobytes (or kilofiles). That is, it multiplies the limit by 1,024. Similarly, "M" denotes megabytes (or megafiles) and "G" denotes gigabytes (or gigafiles). The default for the disk limit is kilobytes.

SEE ALSO

qtree, quota, rquotad

NAME

rc - system initialization command script

SYNOPSIS

/etc/rc

DESCRIPTION

The command script **/etc/rc** is invoked automatically during system initialization. Since the filer has no local editor, **/etc/rc** must be edited from an NFS client with root access to **/etc**. Alternately, you can use the **setup** command to generate a new **/etc/rc** file without using NFS.

EXAMPLE

This is a sample **/etc/rc** file as generated by **setup**:

```
#Auto-generated by setup Tue Jun 2 21:23:52 GMT 1994
hostname filer.dell.com
ifconfig e0 'hostname'-0
ifconfig e1a 'hostname'-1
route add default MyRouterBox 1
routed on
timezone US/Central
savecore
exportfs -a
nfs on
```

FILES

/etc/rc

SEE ALSO

exportfs, exports, hostname, hosts, ifconfig, nfs, route, routed, savecore, setup, timezone, autosupport

NAME

resolv.conf - configuration file for domain name system resolver

SYNOPSIS

/etc/resolv.conf

DESCRIPTION

The resolver configuration file contains information that is read by the resolver routines. The file is designed to be human readable and contains a list of keywords with values that provide various types of resolver information.

The different configuration options are:

nameserver *address*

This specifies the Internet address (in dot notation) of a name server that the resolver should query. Up to 3 name servers may be listed, one per keyword. If there are multiple servers, the resolver queries them in the order listed. When a query to a name server on the list times out, the resolver will move to the next one until it gets to the bottom of the list. It will then restart from the top retrying all the name servers until a maximum number of retries are made.

search *domain-list*

This specifies the search list for host-name lookup. The search list is normally determined from the local domain name; by default, it begins with the local domain name, then successive parent domains that have at least two components in their names. This may be changed by listing the desired domain search path following the **search** keyword with spaces or tabs separating the names. Most resolver queries will be attempted using each component of the search path in turn until a match is found. Note that this process may be slow and will generate a lot of network traffic if the servers for the listed domains are not local, and that queries will time out if no server is available for one of the domains.

The search list is currently limited to six domains with a total of 256 characters.

The keyword and value must appear on a single line, and the keyword (e.g. **nameserver**) must start the line. The value follows the keyword, separated by white space.

FILES

/etc/resolv.conf

SEE ALSO

setup, rc

NAME

rmtab - remote mounted file system table

SYNOPSIS

/etc/rmtab

DESCRIPTION

/etc/rmtab maintains the list of client mount points between server reboots. The list of client mount points can be obtained by using the **MOUNTPROC_DUMP** remote procedure call, or by using the UNIX **showmount** command. When the server successfully executes a mount request from a client, the server appends a new entry to the file. When the client issues an unmount request, the corresponding entry is marked as unused. When the server reboots, unused entries are deleted from the file.

NAME

serialnum - system serial number file

SYNOPSIS

/etc/serialnum

DESCRIPTION

The file **/etc/serialnum** should contain the serial number of your machine. The serial number is found on the back of the machine in the lower right hand corner.

If the file does not exist on your system, create it and put the machine's serial number in it. The file should contain a single line that only has the serial number.

FILES

/etc/serialnum

NAME

shadow - shadow password file

SYNOPSIS

/etc/shadow

DESCRIPTION

The **shadow** file provides more secure storage for the user's password (which would otherwise be in **/etc/passwd).** When the password field of an entry in **/etc/passwd** is empty, **/etc/shadow** must contain a corresponding entry with the same user name but a non-empty encrypted password.

username:password:

The following list explains the required fields:

username — The user's login name, not more than eight characters.

password — The user's password, in an encrypted form that is generated by the UNIX passwd function.

There can be other fields in the **/etc/shadow** file following the ":" after the **password.**

EXAMPLE

Here is a sample **shadow password** file entry:

dave:Qs5l6pBb2rJDA:

SEE ALSO

passwd, options, nis, nsswitch.conf

NAME

sm - network status monitor directory

SYNOPSIS

/etc/sm

DESCRIPTION

The network status monitor provides information about the status of network hosts to clients such as the network lock manager. The network status monitor keeps its information in the **/etc/sm** directory.

The **/etc/sm/state** file contains an integer that is incremented each time the filer is booted.

The **/etc/sm/monitor** file contains a list of network hosts the filer is monitoring.

The **/etc/sm/notify** file contains a list of network hosts that made an NLM lock request to the filer. Each time the filer reboots, it tries to notify the hosts of its new state information. You can remove this file if you want the filer to stop notifying the hosts in this file.

snap sched [*vol_name* [*weeks* [*days* [*hours* [@*list*]]]]]
sets the schedule for automatic snapshot creation. The argument *vol_name* identifies the volume the schedule should be applied to. The second argument indicates how many weekly snapshots should be kept on-line, the third how many daily, and the fourth how many hourly. If an argument is left off, or set to zero, then no snapshot of the corresponding type is created. Daily snapshots are created at 24:00 of each day except Sunday, and weekly snapshots are created at 24:00 on Sunday. Only one snapshot is created at a time. If a weekly snapshot is being created, for instance, no daily or hourly snapshot will be created even if one would otherwise be scheduled. For example, the command

snap sched vol0 2 6

indicates that two weekly snapshots and six daily snapshots of volume *vol0* should be kept on line. No hourly snapshots will be created. For snapshots created on the hour, an optional list of times can be included, indicating the hours on which snapshots should occur. For example the command

snap sched vol0 2 6 8@8,12,16,20

indicates that in addition to the weekly and daily snapshots, eight hourly snapshots should be kept on line, and that they should be created at 8 am, 12 am, 4 pm, and 8 pm. Hours must be specified in 24-hour notation.

With no argument, **snap sched** prints the current snapshot schedule for all volumes in the system. With just the *vol_name* argument, it prints the schedule for the specified volume.

snap reserve [*vol_name*] | [*vol_name percent*] Sets the size of the indicated volume's snapshot reserve to *percent*. With no *percent* argument, prints the percentage of disk space that is reserved for snapshots in the indicated volume. With no argument, the **snap reserve** command prints the percentage of disk space reserved for snapshots for each of the volumes in the system. Reserve space can be used only by snapshots and not by the active file system.

SEE ALSO

df

NAME

snapmirror.allow - list of filers to which you can replicate volumes from this filer

SYNOPSIS

/etc/snapmirror.allow

DESCRIPTION

The **/etc/snapmirror.allow** file exists on the source filer used for SnapMirror. It contains a list of filers to which you can replicate volumes from the source filer. If the source volume and the mirror exist on the same filer, you still must enter the filer name in this file.

In this file, type one filer name per line.

EXAMPLE

The following **snapmirror.allow** file on filerA allows both **filerB** and **filerC** to replicate volumes from filerA:

filerB
filerC

NAME

snapmirror.conf - configuration file specifying how filers replicate volumes using SnapMirror

SYNOPSIS

/etc/snapmirror.conf

DESCRIPTION

The **/etc/snapmirror.conf** file exists on the filer containing the mirror used for SnapMirror. Each entry of the file specifies the volume to be replicated, an argument for the replication, and the schedule for updating the mirror.

Each entry of the **/etc/snapmirror.conf** file is in this format:

source_filer:*source_vol destination_filer*:*destination_vol argument schedule*

The following list describes the fields in each entry:

source_filer the name of the filer containing the source volume.

source_vol the name of the source volume.

destination_filer the name of the filer containing the mirror.

destination_vol the name of the mirror.

argument the maximum speed at which data is transferred, which is specified in kbs (kilobytes per second). Enter a value greater than or equal to 11. By default, the filer transfers the data as fast as it can. To accept the default, specify a dash.

schedule the schedule used by the destination filer for updating the mirror. The schedule contains four fields: minute, hour, day of month, and day of week. The fields are separated from each other by a space. If a field contains more than one value, the values are separated from each other by a comma. A field containing an asterisk means that the field is irrelevant.

EXAMPLE

The following **snapmirror.conf** entry copies **/vol/vol1** on filerA to **/vol/vol2** on filerB at a maximum rate of 2,000 kilobytes per second. FilerB updates the mirror at 10:45 a.m., 11:45 a.m., 12:45 p.m., 1:45 p.m., 2:45 p.m., 3:45 p.m., and 4:45p.m., Monday through Friday. The asterisk means that the data replication schedule is not affected by the day of month.

filerA:vol1 filerB:vol2 kbs=2000 45 10,11,12,13,14,15,16 * 1,2,3,4,5

NAME

syslog.conf - syslogd configuration file

DESCRIPTION

The **syslog.conf** file is the configuration file for the **syslogd** daemon (see **syslogd**). It consists of lines with two TAB separated fields:

selector *action*

The *selector* field specifies the types of messages and priorities to which the line applies. The *action* field specifies the action to be taken if a message the **syslogd** daemon receives matches the selection criteria.

The *selector* field is encoded as a *facility*, a period ("."), and a *level* , with no intervening white-space. Both the *facility* and the *level* are case insensitive.

The *facility* describes the part of the system generating the message, and is one of the following keywords: **auth**, **cron**, **daemon** and **kern**. Here's a short description of each *facil_ity* keyword:

kern Messages generated by the filer kernel.

daemon System daemons, such as the **rshd** daemon (see **rshd**), the routing daemon (see **routed**), the SNMP daemon (see **snmpd**), etc.

auth The authentication system, e.g. messages logged for Telnet sessions.

cron The system's internal cron facility.

The *level* describes the severity of the message, and is a keyword from the following ordered list (higher to lower): **emerg**, **alert**, **crit**, **err**, **warning**, **notice**, **info**, and **debug**.

Here is a short description of each *level* keyword:

emerg A panic condition that results in the disruption of normal service.

alert A condition that should be corrected immediately, such as a failed disk.

crit Critical conditions, such as hard disk errors.

err Errors, such as those resulting from a bad configuration file.

warning Warning messages.

notice Conditions that are not error conditions, but that may require special handling.

info Informational messages, such as the hourly uptime message (see **uptime**).

debug Debug messages used for diagnostic purposes. These messages are suppressed by default.

If a received message matches the specified *facility* and is of the specified *level* (or a higher *level*), the action specified in the *action* field will be taken.

Multiple *selectors* may be specified for a single *action* by separating them with semicolon (";") characters. It is important to note, however, that each *selector* can modify the ones preceding it.

Multiple *facilities* may be specified for a single *level* by separating them with comma (",") characters.

An asterisk ("*") can be used to specify all *facilities* or all *levels*.

The special *level* **none** disables a particular *facility*.

The *action* field of each line specifies the action to be taken when the *selector* field selects a message. There are four forms:

- A pathname (beginning with a leading slash). Selected messages are appended to the specified file.
- A hostname (preceded by an at ("@") sign). Selected messages are forwarded to the **syslogd** daemon on the named host.
- /dev/console. Selected messages are written to the console.
- An asterisk. Selected messages are written to the console.

Blank lines and lines whose first non-blank character is a pound ("#") character are ignored.

It is recommended that all **/etc/syslog.conf** files include the line

***.info /etc/messages**

so that all messages are logged to the /etc/messages file.

EXAMPLES

A configuration file might appear as follows:

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

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

# Also log the messages that go to the console to a remote
# loghost system called adminhost.
*.err;kern.*              @adminhost

# The /etc/secure.message file has restricted access.
auth.notice               /etc/secure.message
```

Also see the sample configuration file in **/etc/syslog.conf.sample**

FILES

/etc/syslog.conf	The **syslogd** configuration file.
/etc/syslog.conf.sample	Sample **syslogd** configuration file.

SEE ALSO

syslogd, messages

NAME

zoneinfo - time zone information files

SYNOPSIS

/etc/zoneinfo

DESCRIPTION

The directory **/etc/zoneinfo** contains time zone information files used by the **timezone** command (see **timezone**). They are in standard UNIX time zone file format as described below.

The time zone information files begin with bytes reserved for future use, followed by six four-byte signed values, written in a "standard" byte order (the high-order byte of the value is written first). These values are, in order:

tzh_ttisgmtcnt The number of GMT/local indicators stored in the file.

tzh_ttisstdcnt The number of standard/wall indicators stored in the file.

tzh_leapcnt The number of leap seconds for which data is stored in the file.

tzh_timecnt The number of "transition times" for which data is stored in the file.

tzh_typecnt The number of "local time types" for which data is stored in the file (must not be zero).

tzh_charcnt The number of characters of "time zone abbreviation strings" stored in the file.

The above header is followed by **tzh_timecnt** four-byte signed values, sorted in ascending order. These values are written in "standard" byte order. Each is used as a transition time at which the rules for computing local time change. Next come **tzh_timecnt** one-byte unsigned values; each one tells which of the different types of "local time" types described in the file is associated with the same-indexed transition time. These values serve as indices into an array of structures that appears next in the file; these structures are written as a four-byte signed **tt_gmtoff** member in a standard byte order, followed by a one-byte signed **tt_isdst** member and a one-byte unsigned **tt_abbrind** member. In each structure, **tt_gmtoff** gives the number of seconds to be added to GMT, **tt_isdst** tells whether this time is during a Daylight Savings Time period and **tt_abbrind** serves as an index into the array of time zone abbreviation characters that follow the structure(s) in the file.

Then there are **tzh_leapcnt** pairs of four-byte values, written in standard byte order; the first value of each pair gives the time at which a leap second occurs; the second gives the *total* number of leap seconds to be applied after the given time. The pairs of values are sorted in ascending order by time.

Then there are **tzh_ttisstdcnt** standard/wall indicators, each stored as a one-byte value; they tell whether the transition times associated with local time types were specified as standard time or wall clock time. A local time transition

specified in standard time ignores any offset due to Daylight Savings Time. On the other hand, a time specified in wall clock time takes the prevailing value of Daylight Savings Time in to account.

Finally there are **tzh_ttisgmtcnt** GMT/local indicators, each stored as a one-byte value; they tell whether the transition times associated with local time types were specified as GMT or local time.

SEE ALSO

timezone

System Services and Daemons

This section contains system services and daemons.

NAME

autosupport - email notification daemon

SYNOPSIS

Data ONTAP 5.3 is capable of sending email notification to other designated addressees in certain situations. The email contains useful information to help them solve or recognize problems quickly and proactively. The system can also be configured to send a short alert notification containing only the reason for the alert to a separate list of recipients. This email is sent only for critical events that might require some corrective action and can be useful for Administrators with alphanumeric pagers that can accept short email messages.

DESCRIPTION

The autosupport mechanism contacts a server system that is listening on the SMTP port (25) to send email. A list of up to 5 mailhosts can be specified and they will be tried in order to send mail out. It sends mail to up to 5 recipient email addresses. The information it sends is described below.

The autosupport mechanism is triggered automatically once a week by the kernel to send information before backing up the messages file. It can also be invoked to send the information through the **options** command. Autosupport mail will also be sent on events that require corrective action from the system administrator. And finally, the autosupport mechanism will send notification upon system reboot from disk.

The subject line of the mail sent by the autosupport mechanism contains a text string to identify the reason for the notification. The messages and other information in the notification should be used to check on the problem being reported. The following are the cases where mail is sent automatically by the system and the subject line text that identifies the reason for the notification. The events that trigger the short note emails (if a recipient list is configured) are noted below and will contain the subject line reason text string and the time of failure in the email data.

1. Weekly notification is marked "WEEKLY_LOG"
2. Data disk failure notification is marked "DISK_FAIL!!!". This event also sends the short note mail.
3. Spare disk failure notification is marked "SPARE_FAIL!!!". This event also sends the short note mail.
4. Disk scrubbing fixing disk errors is marked "DISK_SCRUB!!!".
5. Failure of a fan in the system is notified with "FAN_FAIL!!!". This event also sends the short note mail.
6. Low NVRAM battery triggers notification with "BATTERY_LOW!!!". This event also sends the short note mail.

7. If a disk shelf reports errors, notification is sent with "SHELF_FAULT!!!". This event also sends the short note mail.
8. If one of the power supplies in the filer fails, notification is sent with "POWER_SUPPLY_DEGRADED!!!". This event also sends the short note mail.
9. If the system shuts down because it has detected that the temperature inside the filer is too high, notification is sent with "OVER_TEMPERATURE_SHUTDOWN!!!". This event also sends the short note mail.
10. System reboot notification is sent with "REBOOT". This event also sends the short note mail.

The **setup** command does the following for the autosupport feature:

If an **adminhost** is specified, it adds an entry for **mailhost** with the same address as the **adminhost** to the **/etc/hosts** file.

The following information is sent:

1. Generation date and time stamp
2. Software version
3. System ID
4. Hostname
5. SNMP contact name (if specified)
6. SNMP location (if specified)
7. Output from **sysconfig -v**
8. The system serial number, if the system has one.
9. Currently held license codes
10. Output from **options**
11. Output from **ifconfig -a**
12. Output from **nfsstat -c**
13. Output from **cifs stat**, **cifs sessions**, and **cifs shares**; included if CIFS is licensed.
14. Output from **httpstat**
15. Output from **df**
16. Output from **df -i**
17. Output from **snap sched**

18. Output from **sysconfig -r**
19. The **/etc/messages** file

The autosupport feature is manipulated through the **options** command (see **options**).

The options choices are:

autosupport.enable:
on,**off**

autosupport.mailhost:
Comma-separated list (no spaces)

autosupport.to:
Comma-separated list (no spaces)

autosupport.noteto:
Comma-separated list (no spaces)

autosupport.from:
Local user name

autosupport.doit:
text word describing reason

autosupport.enable: Default is **on**. This option is a switch to enable/disable the autosupport email feature. Customers who wish to disable autosupport permanently will need to set the option in **/etc/rc** with the command

options autosupport.enable off

autosupport.mailhost: Default is **mailhost**. Enter the list of mailhosts separated by "," and no spaces. Up to 5 hosts will be accepted. The autosupport mechanism will try to contact each listed host in turn until it gets a successful SMTP connection. The default mailhost address is set to the **adminhost** address in **/etc/hosts** with a command such as

options autosupport.mailhost mercury,venus,mars

autosupport.to: Enter the list of recipients separated by "," and no spaces. Up to 5 email addresses may be listed with a command such as

options autosupport.to sysadm,autosupport@company.com

autosupport.noteto: Default is an empty list (short note will not be sent). Enter the list of recipients separated by "," and no spaces. Up to 5 email addresses may be listed with a command such as

options autosupport.noteto sysadm1@pager.net,sysadm2@pager.net

autosupport.from: Default is **autosupport**. Enter a user name designated as the sender of the autosupport mail. This allows replies to the mail to be received by a responsible representative at the site.

options autosupport.from sysadm

autosupport.doit: This is a trigger to send the email out. The text word argument to this option is sent in the email subject line. This is used to identify the reason for the notification. To send system information at any time on a running system you can type a command such as

options autosupport.doit SYSTEM_INFO

Error conditions are logged through syslog at level LOG_ERR.

SEE ALSO

options, setup, hosts, rc

NOTES

The autosupport mechanism is enabled by default. When the system boots it will enable the feature. If, for security or other reasons, you wish to disable this feature you should add a line in **/etc/rc** to disable it:

options autosupport.enable off

If you do keep autosupport enabled, remember to add the following lines in **/etc/rc** with your name, phone number and site name. For example,

snmp contact "John - 555-555-1212"
snmp location "Computer Lab"

Add the lines with your information even if you do not use SNMP. This information is sent in the notification and will help Dell support contact you proactively in case of a problem.

NAME

DNS - Domain Name System

DESCRIPTION

Domain Name Service provides information about hosts on a network. This service has two parts: a resolver which requests information and a nameserver which provides it.

Data ONTAP 5.3 supports only the resolver. When the filer needs to resolve a host address, it first looks at the **/etc/nsswitch.conf** (see **nsswitch.conf**) file to get the order in which various name services are to be consulted. If the name services before DNS fail in their lookup and DNS is enabled, then the DNS name server is contacted for address resolution.

DNS can be enabled on the filer by running the **setup** command (see **setup**) or by manually editing the configuration files as described below. If DNS is enabled by running the **setup** command, then the DNS domain name needs to be entered.

Enabling DNS without the setup command:

1. Create the **/etc/resolv.conf** file (see **resolv.conf**) with up to 3 nameservers. Each line contains the keyword **nameserver** followed by the IP address of the server. For example:

   nameserver 192.9.200.1
   nameserver 192.9.201.1
   nameserver 192.9.202.1

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

   options dns.domainname company.com
   options dns.enable on

3. Reboot the filer for these changes to take effect. If the above options commands are also entered from the console, the reboot can be avoided.

Enabling DNS with the setup command:

At setup time, one can choose to enable DNS when prompted to do so. **setup** then queries for the Internet addresses of up to three DNS nameservers.

SEE ALSO

setup, rc, resolv.conf

NAME

NIS - NIS client service

DESCRIPTION

The NIS client service provides information about hosts, user passwords, user groups and netgroups on a network. In NIS terminology, each of the above is referred to as the map and the specific information being looked up is called the key. For example, the hosts map is like the **/etc/hosts** file; it provides a translation from host names to IP addresses. The NIS service typically has two parts: a client component which requests information and a name server which provides it.

Data ONTAP 5.3 supports only the NIS client. When the filer needs to resolve a key in a given map, it looks at the **/etc/nsswitch.conf** (see **nsswitch.conf**) file to figure out the order in which the various databases should be consulted. For example, in case of the hosts map the lookup order may be **file, nis, dns**. This means that the filer will first consult the **/etc/hosts** file. If the host name is not found in the local file, it will then try the NIS service. If the host name is still not found, then it will attempt a DNS lookup.

The NIS client can be enabled on the filer by running the **setup** command (see **setup**) or by manually editing the configuration files as described below. If NIS is enabled by running the **setup** command, then the NIS domain name needs to be entered.

Enabling NIS without the setup command:

1. Edit the **/etc/rc** file (see **rc**) to make sure that the option specifying the NIS domain name is set and the option to enable NIS is on. For example:

   options nis.domainname dell.com
   options nis.enable on

2. Reboot the filer for these changes to take effect. If the above options commands are also entered from the console, the reboot can be avoided. If the options are entered via the console only, they are not saved across a reboot.

Enabling NIS with the setup command:

At setup time, one can choose to enable NIS when prompted to do so. **setup** then queries for the NIS domain name.

SEE ALSO

setup, rc, resolv.conf, nsswitch.conf

NAME

rmt - remote magtape protocol module

SYNOPSIS

/etc/rmt

DESCRIPTION

/etc/rmt is a special command that can be used by remote computers to manipulate a magnetic tape drive over a network connection; for example, the UNIX **dump** and **restore** commands often can either use **/etc/rmt** to access a remote tape, or have **rdump** and **rrestore** variants that can do so. **/etc/rmt** is normally run by the **rshd** daemon (see **rshd**) as a result of a remote machine making a request to **rshd** to do so.

The **/etc/rmt** command accepts requests specific to the manipulation of magnetic tapes, performs the commands, then responds with a status indication. This protocol is provided by **rmt** commands on many UNIX systems, although UNIX systems may support more commands and may give more different error codes.

All responses are in ASCII and in one of two forms. Successful commands have responses of:

A*number***\n**

number is an ASCII representation of a decimal number. Unsuccessful commands are responded to with:

E*error-number***\n***error-message***\n**

error-number is one of:

2 (ENOENT)
The tape device specified in an open request did not have a valid syntax.

6 (ENXIO)
The tape device specified in an open request does not exist.

5 (EIO)
An I/O error occurred when performing the request.

25 (ENOTTY)
An invalid tape operation was specified in a "perform special tape operation" request.

error-message is a (UNIX-style) error string for the error specified by *error-number*.

The protocol is comprised of the following commands, which are sent as indicated - no spaces are supplied between the command and its arguments, or between its arguments, and **\n** indicates that a newline should be supplied:

O*device***\n***mode***\n**
Open the specified *device* using the indicated *mode*. *device* is a tape name of the form described in **tape** and *mode* is an ASCII representation of a decimal number specifying how the tape is to be opened:

0 read-only

1 write-only

2 read-write

If a device had already been opened, it is closed before a new open is performed.

C*device***\n**
Close the currently open device. The *device* specified is ignored.

L*whence***\n***offset***\n**
Performs no operation, and returns the value of *offset*; UNIX-style **lseek** operations are ignored on Dell filer tape devices, just as they are on tape devices on many UNIX systems.

W*count***\n**
Write data onto the open device. If *count* exceeds the maximum data buffer size (64 kilobytes), it is truncated to that size. **/etc/rmt** then reads *count* bytes from the connection, aborting if a premature end-of-file is encountered. The response value is the number of bytes written if the write succeeds, or -1 if the write fails.

R*count***\n**
Read *count* bytes of data from the open device. If *count* exceeds the maximum data buffer size (64 kilobytes), it is truncated to that size. **/etc/rmt** then attempts to read *count* bytes from the tape and responds with **A***count-read***\n** if the read was successful; otherwise an error in the standard format is returned. If the read was successful, the data read is then sent.

I*operation***\n***count***\n**
Perform a special tape operation on the open device using the specified parameters. The parameters are interpreted as ASCII representations of the decimal values. *operation* is one of:

0 write end-of-file marker

1 forward space *count* files

2 backward space *count* files

3 forward space *count* tape blocks

4 backward space *count* tape blocks

5 rewind the tape

6 rewind and unload the tape

The return value is the *count* parameter when the operation is successful.

Any other command causes **/etc/rmt** to close the connection.

DIAGNOSTICS

All responses are of the form described above.

SEE ALSO

tape, rshd

NAME

rquotad - remote quota server

DESCRIPTION

The filer supports the remote quota service that allows NFS clients to determine their quota allocation on the server.

SEE ALSO

quota

NAME

rshd - remote shell daemon

DESCRIPTION

The filer has UNIX-compatible remote shell capability that enables you to execute certain filer commands from a UNIX command line or shell script. It also enables you to use a remote shell application on a PC to enter filer commands.

The **/etc/hosts.equiv** file controls which hosts have access to the filer remote shell. The hosts listed in the **/etc/hosts.equiv** file are called trusted hosts. The filer accepts commands from the filer's administrative users only if the commands are entered through a remote shell.

To see a list of filer commands that can be executed through **rsh**, enter the **rsh ?** command on the trusted host.

EXAMPLE

The following example shows how to enter the **version** command from a trusted host named "adminhost" through a remote shell:

adminhost% **rsh -l root filer version**

SEE ALSO

hosts.equiv, **useradmin**

NAME

snmpd - snmp agent daemon

DESCRIPTION

The filer supports an SNMP version 1 compatible agent that provides support for both the MIB-II management information base for TCP/IP based internets as well as a Dell Custom MIB.

A number of user configurable options for the SNMP agent can be set and queried from the console using the **snmp** command (see **snmp**).

Due to weak authentication in SNMP version 1, SetRequest commands that allow the remote setting of configuration variables have been disabled.

MIB-II

Under MIB-II, information is accessible for the **system, interfaces, at, ip, icmp, tcp, udp** and **snmp** MIB-II groups. The transmission and egp groups are not supported.

The **coldStart, linkDown, linkUp** and **authenticationFailure** traps are implemented. Traps are configured using the **snmp** command.

Dell CUSTOM MIB

The Dell Custom MIB provides a means to obtain detailed information about many aspects of filer operation via SNMP.

The following is a summary of the top-level groups in the Custom MIB and the information they contain:

product
Product-level information such as the software version string and system ID.

sysStat
System-level statistics such as CPU uptime, idle time and aggregate kilobytes received and transmitted on all network interfaces.

nfs
Statistics like those displayed by the **nfsstat** command (see **nfsstat**), including statistics for each client if per-client NFS statistics have been enabled using the **nfs.per_client_stats.enable** option (see **options**). The per-client NFS statistics are indexed by client IP addresses.

quota
Information related to disk quotas, including the output of the quota report command (see **quota**). To access quota information, quotas must be turned on.

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 (see **df**, **maxfiles**, **snap**).

raid

Information on RAID equivalent to the output of the **sysconfig** -**r** command (see **sysconfig**).

SEE ALSO

df, maxfiles, nfsstat, options, quota, snap, snmp, sysconfig

NAME

syslogd - log system messages

DESCRIPTION

The **syslogd** daemon logs system messages to the console, log files and other remote systems as specified by its configuration file, **/etc/syslog.conf**. The **syslogd** daemon reads its configuration file when it starts up during the boot procedure, or within 30 seconds after the **/etc/syslog.conf** file is modified. For information on the format of the configuration file, see **syslog.conf**.

If **/etc/syslog.conf** does not exist the **syslogd** daemon will output all log messages of priority **info** or higher to the console and to the file **/etc/messages**. To prevent **/etc/messages** from getting too large, the **syslogd** daemon will rotate the contents of **/etc/messages** through the files **/etc/messages.0** through **/etc/messages.5**. This rotation is done once a week. So the log messages of the current week will be saved in the file **/etc/messages** and the message logs of the six weeks prior to that are saved in the files **/etc/messages.0** through **/etc/messages.5**.

To prevent large numbers of repeated messages being logged, the **syslogd** daemon will follow the first instance of a repeated message with the number of times the message was repeated. If a message is repeated over a long time period, the **syslogd** daemon will wait for increasingly longer intervals before logging the number of repeats. The repeat notification interval starts at 30 seconds and moves quickly to 20 minutes.

FILES

/etc/syslog.conf	The configuration file.
/etc/syslog.conf.sample	A sample configuration file.
/etc/messages	Message log file for current week.
/etc/messages.[0-5]	Message log for prior weeks.

SEE ALSO

messages, syslog.conf

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.

Address Resolution

The procedure for determining a Media Access Control (MAC) address corresponding to the address of a LAN or WAN destination.

administration host

The client you specify during filer setup for managing the filer. The `setup` program automatically configures the filer to accept `telnet` and `rsh` connections from this client, to give permission to this client for mounting the */* and */home* directories, and to use this client as the mailhost for sending autosupport email messages. At any time after you run the `setup` program, you can configure the filer to work with other clients in the same way as it does with the administration host.

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 e-mail messages from the customer site to a 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.

CIFS

Common Internet File System. A protocol for networking PCs.

client

A computer that shares 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.

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.

filer

A filer is a dedicated, special-purpose network data server that provides fast and reliable file service to network clients connected to Ethernet networks.

GID

Group identification number.

group

A group of users defined in the filer's */etc/group* file.

HTTP

Hypertext Transfer Protocol. An object-oriented protocol that can be used for many tasks, such as name servers and distributed object management systems, through extension of its request methods (commands).

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 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.

network adapter

An Ethernet adapter.

NFS

Network File System. A protocol for networking PCs.

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.

panic

A serious error condition causing the filer to halt; similar to a software crash in the Microsoft® Windows® operating system environment.

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 filers.

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.

qtree

A directory on which you can impose tree quotas, created by the `quota qtree` command.

RAID

Redundant arrays 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).

serial adapter

An expansion card for attaching a terminal as the console on some filers.

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.

SID

Security identifier.

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 enable users to restore files and enable you to back up the filer to tape while the filer is in use.

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.

Unicode

A 16-bit character set standard. It was designed and is maintained by the nonprofit consortium Unicode Inc.

VGA adapter

Expansion card for attaching a VGA terminal as the console.

WAFL

Write Anywhere File Layout. The WAFL™ file system was designed for the Dell™ filer to optimize write performance.

WINS

Windows Internet Name Service.

workgroup

A collection of computers running Microsoft Windows NT® or Windows for Workgroups™ operating systems that is grouped for browsing and sharing.

Index

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

w

Y