Data ONTAP™ 6.4
System Administration
File Access Management Guide

Network Appliance, Inc.
495 East Java Drive
Sunnyvale, CA 94089 USA
Telephone: +1 (408) 822-6000
Fax: +1 (408) 822-4501
Support telephone: +1 (888) 4-NETAPP
Documentation comments: doccomments@netapp.com
Information Web: http://www.netapp.com

Part number 210-00323
January 2003

Copyright and trademark information

Copyright information

Copyright © 1994–2003 Network Appliance, Inc. All rights reserved. Printed in the U.S.A.

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

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

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

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

Software derived from copyrighted material of Network Appliance, Inc. is subject to the following license and disclaimer:

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).

Trademark information

NetApp and the Network Appliance design are registered trademarks of Network Appliance, Inc. in the United States, Canada, and the European Union. Network Appliance is a registered trademark of Network Appliance, Inc. in Monaco and a trademark of Network Appliance, Inc. in the United States and Canada. FAServer is a registered trademark of Network Appliance, Inc. in the United States and the European Union. NetCache is a registered trademark of Network Appliance, Inc. in the European Union and Japan, and a trademark of Network Appliance, Inc. in the United States. SnapCopy is a registered trademark of Network Appliance, Inc. in the European Union and a trademark of Network Appliance, Inc. in the United States. WAFL is a registered trademark of Network Appliance, Inc. in the United States, the European Union, and Canada. FilerView, SecureShare, SnapManager, SnapMirror and SnapRestore are registered trademarks of Network Appliance, Inc. in the United States. Data ONTAP is a trademark of Network Appliance, Inc. in the United States and Canada. Snapshot is a trademark of Network Appliance, Inc. in the United States and the European Union. NetApp–the Network Appliance Company is a registered trademark of Network Appliance, Inc. in the United States and other countries. ApplianceWatch, BareMetal, Center-to-Edge, ContentDirector, DataFabric, MultiStore, NearStore, SecureAdmin, Serving Data by Design, Smart SAN, SnapCache, SnapDrive, SnapVault, SyncMirror, Vfiler, and Web Filer are trademarks of Network Appliance, Inc. in the United States.

Network Appliance is a licensee of the CompactFlash and CF Logo trademarks.

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

Table of Contents

Preface xi

Chapter 1 **Introduction to Network Appliance Filers** 1

About filers 2

How you administer a filer 7

About filer main unit components 9

About disk shelves 11

About Data ONTAP 12

Filer administration with Data ONTAP 16

Getting command information in Data ONTAP 19

Documentation for Data ONTAP 20

Chapter 2 **File Access Using NFS** 23

How resources become available to NFS clients 24

Access restrictions for exported resources 26

How /etc/exports works 29

- Specifying access restrictions in /etc/exports 31
- Specifying root access in /etc/exports 34
- Specifying security services 37
- Exporting resources to netgroups from /etc/exports 39
- Exporting resources to subnets from /etc/exports 41

Exporting or unexporting resources using the exportfs command 43

Providing secure NFS access 47

Enabling and disabling NFS versions 53

Using NFSv4 54

File sharing with (PC)NFS clients 57

- Enabling or disabling the pcnfsd daemon 58
- Creating (PC)NFS user entries on the filer 59

Defining the umask for (PC)NFS-created files and directories 60

Configuring Data ONTAP for WebNFS 61

Chapter 3 **File Access Using CIFS** 63

Filer management through Server Manager and User Manager 65

Displaying and changing the description of the filer 66

Creating NetBIOS aliases for the filer 68

How to specify a Windows name. 70

How CIFS users obtain UNIX credentials 72
- How Data ONTAP maps names 76
- How to specify entries for the /etc/usermap.cfg file 77
- How Data ONTAP interprets domain names in /etc/usermap.cfg 81
- Examples of usermap.cfg entries. 82
- Guidelines and recommendations for mapping users 85
- Mapping a Windows account to root. 87
- Mapping UNIX names to UIDs and GIDs 89
- Migrating CIFS user accounts to the passwd and group files 91
- Creating or disabling the default UNIX user account 92
- Enabling or disabling the Windows guest user account 94

Sharing directories on the filer 95

Displaying and changing share properties 102

Deleting a share. 104

CIFS limitations 105

Understanding local user accounts 108

Creating local groups on the filer. 110

Understanding home directories on the filer 115

How Data ONTAP matches a directory with a user 117

Setting up user home directories 119
- Specifying home directory paths. 120
- Specifying the naming style of home directories 124
- Creating directories in a home directory path 125

Accessing home directories. 127

Specifying support for wide symbolic links in CIFS home directories 130

How to stop offering home shares 131

How share-level access control lists work 132

Specifying how GIDs work with share-level ACLs 133

Changing and displaying a share-level ACL 135

Changing and displaying file-level ACLs 141

Understanding the SID-to-name map cache 144
Enabling and disabling the SID-to-name map cache. 146
Changing the lifetime of SID-to-name mapping entries 147
Clearing the SID-to-name map cache 148

Understanding file screening 150

Enabling and disabling file screening 151

Managing file policies 153

Managing file screening servers 157

Sending a message to all users on a filer 158

Understanding CIFS auditing. 161
Configuring Data ONTAP for CIFS auditing 163
Saving and clearing audit events 165
Understanding event detail displays 173

Understanding oplocks 177

Managing oplocks 179

Displaying CIFS session information 181

Timing out idle sessions 183

Disabling CIFS service 184
Disconnecting selected clients 185
Disabling CIFS for the entire filer 187
Specifying which users receive CIFS shutdown messages 189

Restarting CIFS service. 190

Reconfiguring CIFS for the filer 191

Selecting domain controllers 195

Disabling NetBIOS over TCP 198

Chapter 4 **File Access Using FTP**. 199

Understanding FTP service on the filer. 200

Setting up FTP service on your filer 205
Enabling the FTP service on your filer. 206
Configuring an authentication style 207
Configuring anonymous access to your filer. 210

Denying access to the FTP service 211

Viewing FTP log files 212

Chapter 5 **File Access Using HTTP** 213

Starting the HTTP service 214

Testing the HTTP service. 215

Specifying the filer response to HTTP requests 216
Adding the map rule 217
Adding the redirect rule 219
Adding the pass rule 221
Adding the fail rule. 223

Configuring MIME Content-Type values 224

Maintaining HTTP service security 226
Using HTTP options to restrict access 227
Using an HTTP virtual firewall 228
Protecting Web pages 229
Editing the /etc/httpd.access file 231
Creating and editing httpd.passwd and httpd.group files 234

Using virtual hosting 236

Displaying HTTP statistics 238

Viewing HTTP connection information 242

Chapter 6 **File Access Using DAFS**. 245

Understanding the DAFS protocol 246

Understanding the DAFS service implementation 248

Configuring the DAFS service 249

Starting and stopping the DAFS service 253

Attaching and detaching network devices 257

Understanding session authentication 260

Managing exports for DAFS service 265

Managing client host access to DAFS exports 270

Measuring DAFS performance 275

Displaying DAFS status 280

Configuring DAFS with options 289

Chapter 7 **File Sharing Between NFS and CIFS** 293

Understanding NFS and CIFS file naming 294

Understanding file locking between protocols 296

Understanding read-only bits 297

Controlling CIFS access to symbolic links 299
Understanding how CIFS clients interact with symbolic links 300
About Map entries 303
About Widelink entries. 304
About disabling share boundary checking for symbolic links 306
Redirecting absolute symbolic links 308
Preventing access by CIFS clients to cyclic directory structures 311

Optimizing NFS directory access for CIFS clients 312

Preventing CIFS clients from creating uppercase file names 314

Accessing CIFS files from NFS clients. 315
Adding mapping entries to the WAFL credential cache 316
Deleting mapping entries from the WAFL credential cache. 317
Setting how long mapping entries are valid 318
Monitoring WAFL credential cache statistics 319
Managing mapping inconsistencies 321

Installing SecureShare Access 324

Changing UNIX permissions and DOS attributes from Windows 325

Glossary 329

Index 339

Preface

Introduction

This guide describes how to configure, operate, and manage Network Appliance™ filers that run Data ONTAP™ 6.4 software and are Serving Data By Design™. It covers all filer models.

Audience

This guide is for system administrators who are familiar with operating systems such as UNIX®, Windows 95™, Windows NT®, and Windows 2000®, that run on the filer's clients. It also assumes that you are familiar with how to configure the filer and how Network File System (NFS), Common Internet File System (CIFS), and Hypertext Transport Protocol (HTTP) are used for file sharing or transfers. This guide doesn't cover basic system or network administration topics, such as IP addressing, routing, and network topology; it emphasizes the characteristics of the NetApp® filer.

Terminology

When the term "filer" is used, the information applies to all NetApp filer models, including the FAS900 and R100 series of storage appliances.

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.

Command conventions

You can enter filer commands either on the system console or from any client computer that can access the filer through a Telnet session.

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.

Keyboard conventions

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

Typographic conventions

The following table describes typographic conventions used in this guide.

Convention	Type of information
Italic font	Words or characters that require special attention. 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. 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.

Introduction to Network Appliance Filers 1

About this chapter

This chapter describes what a filer is, what it consists of, and how you can use the filer software, Data ONTAP, to configure and manage filers.

Topics in this chapter

This chapter discusses the following topics:

- "About filers" on page 2
- "How you administer a filer" on page 7
- "About filer main unit components" on page 9
- "About disk shelves" on page 11
- "About Data ONTAP" on page 12
- "Filer administration with Data ONTAP" on page 16
- "Getting command information in Data ONTAP" on page 19
- "Documentation for Data ONTAP" on page 20

About filers

What a filer is

A filer is a hardware and software storage system that serves data. It acts on network requests from clients and processes them by writing data to or retrieving data from disks in disk shelves that are connected to it. The software that enables the filer to perform these tasks is the Data ONTAP operating system.

Components of a filer

A filer consists of the following major components.

Component	Function	To learn more, read...
Filer main unit, or chassis	The piece of hardware that receives and sends data	"About filer main unit components" on page 9
Disk shelves	Hardware that holds disks and is connected to the filer	"About disk shelves" on page 11
Data ONTAP	Software that is the operating system for the filer	"About Data ONTAP" on page 12

What a filer does

A filer provides the following features.

Feature	Description
Network file service	The filer enables users on client workstations (or hosts) to create, delete, modify, and access files or blocks stored on it. Client workstations are connected to the filer through TCP/IP, direct-attached, or fabric-attached network connections.

Feature	Description
Multiprotocol file and block sharing	Clients can use the following protocols to access data on the filer: ◆ CIFS (Common Internet File System)—used by Windows® clients ◆ FCP (Fibre Channel Protocol)—used for block access in storage area networks ◆ DAFS (Direct Access File System)—used by DAFS clients ◆ HTTP (HyperText Transmission Protocol)—used by the World Wide Web ◆ iSCSI (Internet small computer system interface)—used for block access in storage area networks ◆ NFS (Network File System)—used by UNIX systems ◆ (PC)NFS (Personal Computer NFS)—used by PCs to access NFS 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. Blocks written using one protocol can also be accessed by clients using the other protocol.

Feature	Description
Data protection	The filer protects disk data in the following ways: Snapshot™ same-volume backup enables you to manually or automatically create, schedule, and maintain multiple backups (or snapshots) of volume and qtree images using only a minimal amount of additional volume space at no performance cost. SnapMirror® (license required) filer-to-filer snapshot mirroring enables you to mirror snapshots on one filer to a partner filer. Should the original filer be disabled, this ensures quick restoration of data from the point of the last snapshot.

Feature	Description
Data protection (continued)	SnapVault™ (license required) multiple sources-to-central system snapshot backup backs up snapshots from qtrees on multiple volumes and filers to a single Network Appliance SnapVault secondary storage system. This system is specialized for quick backup and restore of its filer sources. The SnapRestore® (license required) feature performs fast restoration of backed up data on request from snapshots on an entire volume. Tape backup dump and restore commands enable you to back up filer or SnapVault snapshots to tape. Because the snapshot rather than the active file system is backed up to tape, the filer can continue its normal functions while the tape backup is still in operation. Vol copy (license required) fast block copy of data enables you to quickly copy blocks of stored filer data from one volume to another. SyncMirror™ (cluster configuration required) performs real-time mirroring of data to two separate filer volumes physically connected to the same filer head. In case of an unrecoverable disk error on one volume, the filer automatically switches access to the mirrored volume. NDMP (Network Data Management Protocol) support enables third-party applications that use NDMP to manage tape backup operations of filer data. The NDMP copy command carries out NDMP-compliant backups and restores. Security login restricts access to NDMP operations. The `nvfail` option provides protection against data corruption by nonvolatile RAM (NVRAM) failures.

Feature	Description
Autosupport	Mail notifications about filer problems are automatically sent to Network Appliance Technical Support.

How you administer a filer

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

- Command execution through the filer's command line
- Command execution through Windows NT and Windows 2000
- Configuration file editing
- Command execution through the FilerView® interface
- Command execution through DataFabric™ Manager

Command execution through the filer's command line

You use the filer's command line to execute all Data ONTAP 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 and Windows 2000

You use Windows NT and Windows 2000 commands to perform filer administrative tasks related to Windows NT and Windows 2000.

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 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 main unit of the filer has two kinds of components: internal components that enable it to function, and slots and ports that connect it to networks and disk shelves.

Internal filer components

The following internal components enable the filer to function.

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 RAM)	Data ONTAP uses NVRAM to log network transactions as a data integrity measure. In case of a system or power failure, Data ONTAP uses the contents of NVRAM to restore network data to disk.
Diskette drive (not available on all models)	In an emergency and for major upgrades, Data ONTAP uses the diskette drive to boot from diskettes.
CompactFlash™ card (not available on all models)	The filer automatically boots from a Data ONTAP release stored on the CompactFlash card. The CompactFlash card also stores a backup version of Data ONTAP from which to boot the filer in an emergency.

Component	Description
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.

Component	Description
Slots	The filer contains expansion slots for network interface cards, tape drive adapters, and disk shelf adapters.
Serial ports	The two serial ports are as follows: ◆ The console port connects the filer to a serial terminal that you can use as a console. ◆ The diagnostics port connects diagnostic equipment such as the environmental monitor unit (EMU) of a StorageShelf 2 storage shelf.

About disk shelves

Disk shelves contain disks

A disk shelf contains the disks that store the data that the filer serves.

Disk shelf environmental information

Disk shelves collect information about the presence of disks, fan status, power supply status, and temperature. Disk shelves send messages to the console if parameters exceed permissible operating conditions.

About Data ONTAP

Data ONTAP overview

You administer the filer using Data ONTAP commands. Data ONTAP uses only commands and files that are relevant to file service administration. Data ONTAP contains no extraneous utilities such as text editors.

Data ONTAP manages data in four ways:

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

Data storage management

Data ONTAP stores data on disks in disk shelves. Disks are organized into RAID groups, and RAID groups are organized into volumes. These items are explained in the following table, along with what aspects of them you can administer.

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, up to 52 data disks before Data ONTAP 5.0 and 28 data disks in Data ONTAP 5.0 and later, and optional spare disks. Data ONTAP supports up to 150 RAID groups per filer or cluster.	You can control the size of a RAID group. This enables you to customize backups and disk failure recovery.

Data storage concept	Explanation	Storage administrative actions
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 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 from 1 to 200 volumes.	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 you flexibility in managing data storage tasks such as backups and restores, and enables you to customize volumes.
Logical unit numbers (LUNs)	Logical units of storage, each of which is identified by a number, used in a storage area network (SAN) environment.	You can create LUNs within volumes and specify the size of a LUN.

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 the following table.

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, files that use NT file system (NTFS) security, or both types 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 to customize areas for projects and to keep users and projects from monopolizing resources.

Data access management

Data ONTAP 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 enables you to perform the following actions:

- Manage network connections
- Manage protocols
- Configure pairs of filers into clusters for failover
- Dump data to tape and restore it to the filer
- Copy volumes
- Mirror volumes (synchronously and asynchronously)
- Create vfiler™ units

Data migration management

Data ONTAP manages data migration by means of

- Snapshots
- Asynchronous mirroring
- Synchronous mirroring
- Backup to tape
- Volume copy

Filer administration with Data ONTAP

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 CIFS access so that you can identify potential security problems

You maintain client access by the following methods:

- Configuring volumes and qtrees to accommodate the needs of Windows 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

The following table 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.

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
	Review DAFS utilization, if licensed for DAFS.	`dafs stats` command

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 disk shelves are connected properly.	`shelfchk` command
	Review filer uptime.	`uptime` command
	Review filer OS revision.	`version` command
	Review current Network Information Service (NIS) server.	`ypwhich` command

Getting command information in Data ONTAP

Where information about Data ONTAP commands is

Basic information about each Data ONTAP command is stored in manual pages in the etc/man directory in the filer. You can access this information from the command line using the `man` command.

How to get information about Data ONTAP commands

To get information about a Data ONTAP command, complete the following step.

Step	Action
1	Enter the following command: **`man [-s section] command_name`** *section* is one of the four sections of Data ONTAP man page information. Section 1 is Commands, Section 4 is Special Files, Section 5 is File Formats and Conventions, and Section 8 is System Management and Services. **Result:** The manual page information is displayed on the terminal.

Documentation for Data ONTAP

Description of Data ONTAP documents

The following table describes the documents in the Data ONTAP documentation library.

Document name	Information included
SAN Setup Overview for FCP	Describes how to configure a new filer in a SAN environment for the first time using FCP.
SAN Setup Overview for iSCSI	Describes how to configure a new filer in a SAN environment for the first time using iSCSI.
Software Setup Guide	Describes how to set up and configure Network Appliance filers and NearStore™ R100 series systems that run Data ONTAP software.
MultiStore™ Management Guide	Describes how to use the MultiStore software product on Network Appliance filers that run Data ONTAP.
Network Management Guide	Describes how to configure and manage network interfaces, virtual network interfaces (vifs), virtual local area networks (VLANs), and routing on Network Appliance filers that run Data ONTAP software.
Data Protection Guide	Describes how to back up, restore, and copy data between Network Appliance filers that run Data ONTAP software. It covers all filer models.

Document name	Information included
Storage Management Guide	Describes how to configure, operate, and manage Network Appliance storage appliances, such as filers, that run Data ONTAP software.
File Access Management Guide	Describes how to configure and manage file access using the CIFS, NFS, HTTP, FTP, and DAFS protocols.
Block Access Management Guide	Describes how to manage a filer in a SAN configuration.
Upgrade Guide	Describes how to upgrade Network Appliance filers to the latest release of Data ONTAP firmware and software.
Commands: Quick Reference	Lists Data ONTAP commands and briefly summarizes what they do.
Commands: Manual Page Reference	Contains man pages for Data ONTAP commands, describing their use in detail.

File Access Using NFS 2

About this chapter

This chapter provides information about preparing your filer for sharing resources with NFS and (PC)NFS clients. A resource is a file, directory, or volume on the filer. This chapter also discusses how the filer transfers files to WebNFS clients.

Topics in this chapter

This chapter discusses the following topics:

- "How resources become available to NFS clients" on page 24
- "Access restrictions for exported resources" on page 26
- "How /etc/exports works" on page 29
- "Exporting or unexporting resources using the exportfs command" on page 43
- "Providing secure NFS access" on page 47
- "Enabling and disabling NFS versions" on page 53
- "Using NFSv4" on page 54
- "File sharing with (PC)NFS clients" on page 57
- "Defining the umask for (PC)NFS-created files and directories" on page 60
- "Configuring Data ONTAP for WebNFS" on page 61

How resources become available to NFS clients

Resources that NFS clients can mount

NFS clients can mount only those resources that have been exported. An exported resource is one that you have made available for mounting.

How Data ONTAP matches mount requests with exported resources

When Data ONTAP receives a mount request from a client, it compares the path name in the mount request to the path names of the exported resources. If Data ONTAP finds a match, the NFS client can mount the resource. Data ONTAP uses the longest matching prefix in the path name in determining access restrictions.

Example: Suppose you export both the /vol/vol0/home resource and the /vol/vol0 resource. When a host tries to mount the /vol/vol0/home/user1 directory, Data ONTAP applies the access restrictions specified for the /vol/vol0/home resource to determine whether the host should be granted access. Similarly, when a host tries to mount the /vol/vol0 directory, Data ONTAP applies the access restrictions specified for the /vol/vol0 resource.

Ways to export resources

You can export resources in two ways:

- Specify resources in the /etc/exports file and execute the `exportfs` command either at the command line or from the /etc/rc file to export the resources according to the specifications in the /etc/exports file.

  This method is for exporting resources that you want to export after each filer reboot.

- Use the `exportfs` command to export resources that are not specified in the /etc/exports file or to export resources specified in the /etc/exports file with different access restrictions.

  This method is for exporting resources temporarily because the `exportfs` command is not persistent across filer reboots.

For information about specifying what and how resources are exported in the /etc/exports file, see "How /etc/exports works" on page 29. For information about how to use the `exportfs` command, see "Exporting or unexporting resources using the exportfs command" on page 43.

Rules for exporting resources

Regardless of the method for exporting a resource, remember these rules:

- You must specify the complete path name for a resource to be exported. That is, the path name must begin with the /vol prefix.
- You cannot export /vol, which is not a path name to a file, directory, or volume. If you want to export all volumes on the filer, you must export each volume separately.
- When you export a resource to multiple targets, separate the target names with a colon (:).

About exporting different levels of a directory hierarchy

The way Data ONTAP exports a resource is similar to that used by the SunOS 4.x operating system. However, Data ONTAP enables you to export a resource whose ancestor has already been exported. For example, you can export the /vol/vol0/home/users directory after you export the /vol/vol0/home directory.

When exporting different levels of a directory hierarchy, take care not to give clients greater access permissions at a higher level in the hierarchy. For example, in the following /etc/exports file, the first entry specifies no access restrictions for the /vol/vol0 volume. That is, any host can mount the /vol/vol0 volume as read-write. The second entry restricts access to the /vol/vol0/home directory to specified hosts. However, because the first entry gives any host permission to mount the /vol/vol0 volume, hosts other than host1 and host2 can have access to the /vol/vol0/home directory by mounting the /vol/vol0 volume.

```
/vol/vol0

/vol/vol0/home -access=host1:host2
```

Access restrictions for exported resources

About access restrictions

When you export a resource, you can specify the access restrictions that govern how the resource can be mounted. If you export a resource without specifying access restrictions, it can be mounted read-write by all hosts.

You specify access restrictions with options in the /etc/exports file or in the `exportfs` command. These access restrictions options determine

- Which hosts can mount the resource
- Whether the resource can be mounted read-write or read-only
- Whether the root user on the client can access the resource
- The user ID (UID) of the user accessing the resource

Meanings of access restriction options

The access option: It lists individual hosts, netgroups, or subnets that can mount the resource. Hosts not included in the access list are denied access to the resource. By default, the listed hosts can mount the resource read-write.

There is no limit to the number of targets you can list with the `access` option in the /etc/exports file. However, if you want to avoid exceeding the maximum length of an /etc/exports file entry, use netgroups or subnet addresses instead of individual host names in the list.

The root option: It determines the user ID for the root user on the client who sends the NFS requests after mounting the resource. The following list describes the effect of including or excluding a host from the `root` option:

- If you specify a host with the `root` option, the root user on that host keeps the root UID (0) when accessing the resource.

Note

Having a root UID does not guarantee read-write access. If an `ro` option is specified for a resource, the `ro` option overrides the `root` option, making the resource a read-only resource.

- If no `root` option is specified for the resource, or if the host trying to access the resource is not specified with the `root` option, access by root on the client is either denied or subject to modification by the `anon` option.

You can specify up to 255 names with the `root` option. You can specify host names and subnets; you cannot specify netgroups.

The rw option: It gives read-write access to the specified hosts. The `rw` option is useful if you want to make a resource readable by most hosts.

You can specify up to 255 names with the `rw` option. The names can be host names or subnets; you cannot specify netgroups.

The ro option: It exports the resource read-only.

This option is not followed by a list of targets because it applies to all hosts that have permission to mount the resource.

The anon option: It determines the UID for the root user on the client. The following table describes the effects of the `anon` option.

If...	Then...
The client is not specified by the `root` option for the resource	The UID specified by the `anon` option is the effective UID.
The client is specified by the `root` option for the resource	The UID specified by the `anon` option is not used. The effective UID is 0.

The following list describes the default UID specified by the `anon` option and the meanings of some special UIDs:

- By default, the `anon` option specifies a UID of 65534. That is, if you do not use the `root` and `anon` options for a resource, root users on all hosts access the resource using the UID 65534.
- If the `anon` option specifies a UID of 65535, root access is disabled.
- If the `anon` option specifies a UID of 0, root access is granted to all hosts.

Interactions among options

When Data ONTAP receives a mount request for a resource, it takes into consideration all options pertaining to the resource to determine access restrictions.

Example: The /etc/exports file contains the following entry for the /vol/vol0 volume:

```
/vol/vol0 -access=host1:host2,root=host1,rw=host2
```

The `access` and `root` options allow host1 to mount the /vol/vol0 volume with root access. The `rw` option, however, specifies that hosts other than host2 have only read access. As a result, host1 can only mount the /vol/vol0 volume as a read-only resource.

Example: The /etc/exports file contains the following entry for the /vol/vol0 volume:

```
/vol/vol0 -access=host1:host2,root=host1,ro
```

The `access` and `root` options allow host1 to mount the /vol/vol0 volume with root access. The `ro` option, however, makes the /vol/vol0 volume a read-only resource to all hosts. As a result, host1 can only mount the /vol/vol0 volume as a read-only resource.

How /etc/exports works

About /etc/exports

The /etc/exports file contains a list of resources that can be exported. When rebooting, Data ONTAP executes the `exportfs -a` command in the /etc/rc file to automatically export all resources in /etc/exports file. As a result, you need not enter the `exportfs` command to export each resource.

In the /etc/exports file, you also specify the access restriction and security service with which clients can mount each resource. At any time after the filer is up and running, you can override any specification in the /etc/exports file with the `exportfs` command.

When changes to /etc/exports take effect

Entries in the /etc/exports file take effect when one of the following events happens:

- Data ONTAP executes the `exportfs -a` command. This exports all entries in the /etc/exports file.
- You enter the `exportfs` command followed by a path name specified in the /etc/exports file. This exports the specified entry.

For more information about how to use the `exportfs` command, see "Exporting or unexporting resources using the exportfs command" on page 43.

Targets to which you export a resource

You can specify in the /etc/exports file to export a resource to these targets:

- Hosts
- Netgroups
- Subnets

Information in /etc/exports

The /etc/exports file specifies the following types of information:

- Complete path name of each resource to be exported.
- Targets to which each resource is exported. Targets are optional.
- Access restrictions with which each resource can be mounted. Access restrictions are optional.
- Security service each resource uses. Security services are optional.

Rules for specifying an /etc/exports entry

The following rules determine how a resource is exported:

- If the /etc/exports file entry contains only the path name of a resource, that resource can be mounted read-write by all hosts.
- If the /etc/exports file entry contains the path name of a resource followed by options, the resource is exported with access restrictions, which determine
  - Which targets have permission to mount the resource.
  - The type of access allowed for each target (for example, read-only or read-write).
  - How NFS requests made by root on the client are handled. The requests can be denied, allowed, or allowed with a converted UID.
- In an /etc/exports file entry, the first access restriction option is preceded by a dash (-). Subsequent options are preceded by a comma (,).

Example: `/vol/vol0 -access=host1:host2,root=host1,rw=host2`

The steps for exporting a resource from the /etc/exports file vary with the type of target to which you export the resource.

Limitation of the /etc/exports file

Each line of the /etc/exports file can contain up to 1,024 characters.

For detailed information

The following sections describe how different options in the /etc/exports file control the ways resources are exported:

- "Specifying access restrictions in /etc/exports" on page 31
- "Specifying root access in /etc/exports" on page 34
- "Specifying security services" on page 37
- "Exporting resources to netgroups from /etc/exports" on page 39
- "Exporting resources to subnets from /etc/exports" on page 41

Specifying access restrictions in /etc/exports

Specifying the same restrictions for all hosts

For all hosts to mount read-write: To enable all hosts to mount the resource read-write, complete the following step.

Step	Action
1	Enter the path name of the resource without any options. **Example:** `/vol/vol0/home` **Result:** All hosts can mount the /vol/vol0/home directory as read-write.

For all hosts to mount a resource read-only: To enable all hosts to mount the resource read-only, complete the following step.

Step	Action
1	Enter the path name of the resource with the `ro` option. **Example:** `/vol/vol0/home -ro` **Result:** All hosts can mount the /vol/vol0/home directory as read-only.

Specifying different restrictions for different hosts

For some hosts to mount read-write: To have some hosts mount the resource read-write and other hosts be unable to mount the resource, complete the following step.

Step	Action
1	Enter the path name of the resource with the `access` option. **Example:** `/vol/vol0/home -access=host1:host2` **Result:** Host1 and host2 can mount the /vol/vol0/home directory as read-write. Other hosts cannot mount the /vol/vol0/home directory.

For some hosts to mount read-only: To have some hosts mount the resource read-only and other hosts be unable to mount the resource, complete the following step.

Step	Action
1	Enter the path name of the resource with the `access` and `ro` options. **Example:** `/vol/vol0/home -access=host1:host2,ro` **Result:** Host1 and host2 can mount the /vol/vol0/home directory as read-only. Other hosts cannot mount the /vol/vol0/home directory.

For some hosts to mount read-write and all others to mount read-only: To enable some hosts to mount a resource read-write and all other hosts to mount the resource read-only, complete the following step.

Step	Action
1	Enter the path name of the resource with the `access` and `rw` options. **Example:** `/vol/vol0/home -rw=host1:host2` **Result:** Host1 and host2 can mount the /vol/vol0/home directory as read-write. All other hosts can mount the /vol/vol0/home directory as read-only.

For some hosts to mount read-write and some hosts to mount read-only: To have some hosts mount a resource read-write and some hosts mount the resource read-only, complete the following step.

Step	Action
1	Enter the path name of the resource with the `access` and `rw` options. The hosts with the `rw` option must be a subset of the hosts specified with the `access` option. Hosts specified after the `access` option but not after the `rw` option can mount the resource read-only. **Example:** `/vol/vol0/home -access=host1:host2,rw=host1` **Result:** Host1 can mount the /vol/vol0/home directory as read-write. Host2 can mount the /vol/vol0/home directory as read-only. Hosts other than host1 and host2 cannot mount the /vol/vol0/home directory.

Specifying root access in /etc/exports

About root access

After the root user on a client mounts a resource, depending on the specifications in the /etc/exports file, the root user might be

- Able to access the resource with UID 0
- Able to access the resource with a nonroot UID
- Denied access

Specifying the same root access restriction for all clients

For all clients to have root access: To give all clients root access, complete the following step.

Step	Action
1	Enter the path name of the resource with the `anon` option. **Example:** `/vol/vol0/home -anon=0` **Result:** The root user on any client can access the /vol/vol0/home directory with UID 0.

For no clients to have root access: To deny root users on all clients access to the resource, complete the following step.

Step	Action
1	Enter the path name of the resource with the `anon` option. **Example:** `/vol/vol0/home -anon=65535` **Result:** No root users on clients who have mounted the /vol/vol0/home directory can access the directory.

For all clients to be considered nonroot users: To have root users on all clients be considered nonroot users when accessing the resource, complete the following step.

Step	Action
1	Enter the path name of the resource with the `anon` option. **Example:** `/vol/vol0/home -anon=65530` **Note** The UID specified must not be 0. **Result:** All root users on clients who have mounted the /vol/vol0/home directory access the directory with UID 65530.

For root users on all clients to be considered nobody: To have root users on all clients access the resource as nobody, complete the following step.

Step	Action
1	Enter the path name of the resource with the `anon` option. **Example:** `/vol/vol0/home -anon=65534` **Result:** All users on clients who have mounted the /vol/vol0/home directory can access the directory with UID 65534, which is the UID for a special user named nobody.

Specifying that some users have root access

For some clients to have root access and others no root access: To have the root users on some clients retain the root UID and give others no root access, complete the following step.

Step	Action
1	Enter the path name of the resource with the `root` and `anon` options. **Example:** `/vol/vol0/home -root=host1:host2,anon=65535` **Result:** Root users on host1 and host2 access the /vol/vol0/home directory as root and root users on other clients are denied access.

For root users on some clients to have root access and others non-root access: To have users on some clients retain the root UID and give others a nonzero UID, complete the following step.

Step	Action
1	Enter the path name of the resource with the `root` and `anon` options. **Example:** `/vol/vol0/home -root=host1:host2,anon=65530` **Note** The UID specified must not be 0. **Result:** Root users on host1 and host2 access the /vol/vol0/home directory as root and root users from other clients use UID 65530.

For root users on some clients to have root access and others to be treated as nobody: To have users on some clients retain the root UID and others be considered nobody, complete the following step.

Step	Action
1	Enter the path name of the resource with the `root` and `anon` options. **Example:** `/vol/vol0/home -root=host1:host2,anon=65534` **Result:** Root users on host1 and host2 access the /vol/vol0/home directory as root and root users from other clients use UID 65534, the UID for the special user named nobody.

Specifying security services

About security services

Data ONTAP allows you to specify a security service for the resource you are mounting. By specifying the security service, you determine how secure NFS access is to the filer. For information about setting up secure NFS access, see "Providing secure NFS access" on page 47.

Security service types

You can create different levels of secure NFS access by specifying different security service types. Data ONTAP supports two security service types. The two types, in ascending order of strength, are as follows:

- UNIX authentication—Does not use strong cryptography and is the least secure of the security services.
- Kerberos version 5 (v5)—Provides three types of security service.
  - Authentication—Uses strong cryptography to prove a user's identity to a filer and to prove a filer's identity to a user.
  - Integrity—Provides a cryptographic checksum of the data portion of each request and response message in those messages to prove no attacker in the middle has tampered with the NFS traffic.
  - Privacy—Encrypts the contents of packets, including procedure arguments and user data, in both directions using a shared session key established by the client from the filer.

Note

UNIX authentication is the default security service used by Data ONTAP.

Specifying security services

Data ONTAP allows you to specify more than one security service. This enables users who are using different levels of security to have NFS access. If you want users to use a particular level of security, specify only one security service in the /etc/exports file.

To specify more than one security service type, complete the following step.

Step	Action
1	Enter the path name of the resource with the `sec` option. **Example:** `/vol/vol0/home -sec=krb5:krb5i:krb5p` **Result:** Users can use Kerberos v5 authentication, integrity, or privacy services to access the /vol/vol0/home directory.

To specify only one security service type, complete the following step.

Step	Action
1	Enter the path name of the resource with the `sec` option. **Example:** `/vol/vol0/home -sec=krb5p` **Result:** Users can use only Kerberos v5 privacy service to access the /vol/vol0/home directory.

Exporting resources to netgroups from /etc/exports

About netgroups

When exporting a resource, if you do not want to list names of individual hosts as targets, you can specify a predefined netgroup as the target.

You can create netgroups in the /etc/netgroup file on the filer or in the netgroup.byhost NIS map.

Relationship between /etc/netgroup and the netgroup.byhost map

Information in the /etc/netgroup file is not linked to the netgroup.byhost NIS map. For example, the NIS master does not use the /etc/netgroup file as input for building the netgroup.byhost map. You can, however, copy the information from the NIS map to the /etc/netgroup file so that the filer can use the /etc/netgroup file at times when the NIS server is not available.

Restrictions on exporting to netgroups

When exporting to netgroups, remember these restrictions:

- The `access` option is the only export option you can use for a netgroup. You cannot specify a netgroup with the `rw` or `root` option.
- Each line in the /etc/netgroup file can contain up to 4,096 characters.

Format of a netgroup entry

Each entry in the /etc/netgroup file contains the netgroup name and a member list. You can specify one of these items in the member list:

- Another netgroup name
- An entry in the following form:

  (*host_name*, *user_name*, *domain_name*)

  host_name is a fully qualified name if the domain is not the local domain.

  user_name is ignored, so leave this field empty.

  domain_name is the name of the domain in which the netgroup is valid. Use the local domain name or leave this field empty.

  Even if you leave a field empty, you must include the commas in the entry.

Example of a netgroup file

The following /etc/netgroup file contains three netgroups:

```
trusted-hosts (host1,,) (host2,,)
untrusted-hosts (host3,,) (host4,,) (host5,,)
all-hosts trusted-hosts untrusted-hosts
```

In an exports entry, you can specify the trusted-hosts, untrusted-hosts, or all-hosts netgroup as the export target with the `access` option.

Exporting a resource to a netgroup

To export a resource to a netgroup, complete the following steps.

Step	Action	
1	**If...**	**Then...**
	You want the filer to use NIS for authenticating (PC)NFS users	Go to Step 2.
	The filer is not an NIS client or you want to use different netgroup definitions from the ones specified in the netgroup.byhost map	Create the /etc/netgroup file, then go to Step 2.

2	Type the following entry in the /etc/exports file:
	***path_name* -access=*netgroup_name*[:*netgroup_name*:...]**
	Example: The /etc/exports file contains the following entries:
	`/vol/vol0 -access=trusted-hosts,root=host1`
	`/vol/vol0/home -access=untrusted-hosts`
	Result: Hosts that belong to the trusted-hosts netgroup can mount the /vol/vol0 volume as read-write. The root user on host1, which is a member of trusted-hosts, can access the /vol/vol0 volume as root. Hosts that belong to the untrusted-hosts netgroup can mount the /vol/vol0/home directory as read-write.	

Exporting resources to subnets from /etc/exports

Reasons for exporting to a subnet

Exporting to a subnet has the same effect as exporting to individual hosts on that subnet, but it gives you the convenience of not having to list individual host names in the /etc/exports file. If all hosts on the same subnet should mount the same resources, export the resources to the subnet.

Format of the export target

The following table describes the format of the export target if you want to export a resource to a subnet.

Information specified	Format

Network address and number of leading 1 bits in the subnet mask	*dotted_ip*/*num_bits*
	
	dotted_ip is an IP address or an IP subnet.
	
	num_bits is the number of leading 1 bits of the subnet mask.
	
	Examples: If you want to export to IP addresses of the form 123.45.67.x with 24 leading 1 bits in the subnet mask, specify the following export target:
	
	`123.45.67.0/24`
	
	If you want to export to IP addresses of the form 123.45.x.y with 16 leading 1 bits in the subnet mask, specify the following export target:
	
	`123.45.0.0/16`
	
	In these examples, you can omit the “0” from the *dotted_ip* field.

Information specified	Format
Network address and subnet mask	"[network] *dotted_ip* [netmask] *netmask*" *dotted_ip* is an IP address or an IP subnet. *netmask* is the netmask specified with four octets. **Examples:** If you want to export to IP addresses of the form 123.45.67.x with a netmask of 255.255.255.0, use either of these values as the export target: "123.45.67 255.255.255.0" "network 123.45.67 netmask 255.255.255.0"

Exporting a resource to a subnet

To export a resource to a subnet, complete the following step.

Step	Action
1	Type the following entry in the /etc/exports file: ***path_name* -access=*subnet_address*[:*subnet_address*:...]** **Example:** /vol/vol0/home -access=123.45.67/24 **Result:** Hosts with IP addresses in the form of 123.45.67.x can mount the /vol/vol0/home directory as read-write. **Note** You can specify additional access restrictions when exporting to subnets, just as you do when exporting to hosts. See "Specifying access restrictions in /etc/exports" on page 31 for more information about how to specify access restrictions.

Exporting or unexporting resources using the exportfs command

Purposes of the exportfs command

You use the `exportfs` command for these purposes:

- Export entries specified in the /etc/exports file.
- Export entries in different ways from those specified in the /etc/exports file. For example, you can export resources that are not specified in the /etc/exports file or resources that are specified in the /etc/exports file but with different access restrictions.
- Unexport entries that are currently exported. After a resource is unexported, it becomes unavailable for clients to mount.

Rules for specifying multiple access restriction options

In the `exportfs` command, the first access restriction option is preceded by `-o` and a space. Subsequent options are preceded by a comma (,).

Example: `exportfs -o access=host1:host2,root=host1,rw=host2 /vol/vol0`

Targets to which you export a resource

You can specify in the `exportfs` command to export a resource to these targets:

- Hosts.
- Netgroups. See “Exporting resources to netgroups from /etc/exports” on page 39 for information about how to define a netgroup.
- Subnets. See “Exporting resources to subnets from /etc/exports” on page 41 for information about the format of a subnet address.

Exporting entries in /etc/exports

To export entries in the /etc/exports file, complete the following steps.

Step	Action	
1	**If...**	**Then...**
	You want to export all entries in the /etc/exports file	Go to Step 2.
	You edited an entry in the /etc/exports file and you want the change to take effect	Go to Step 3.
2	Enter the following command: **`exportfs -a`**	
3	Enter the following command: **`exportfs path_name`** *path_name* is the path name to a resource specified in the /etc/exports file.	

Note

Network Appliance recommends that you confirm that the exports shown in the output of the `exportfs` command match the exports specified in /etc/exports. If they do not match, a client might get stale file handles after you reboot the filer.

Exporting entries differently than specified in /etc/exports

To export entries not in the /etc/exports file or to export entries with access restrictions that are different from those specified in the /etc/exports file, complete the following step.

Step	Action

1	Enter the following command:
	`exportfs [ -i ] [ -o options ] path_name`
	Omit the `-i` option if you are exporting a resource not included in the /etc/exports file. Use the `-i` option if the resource is included in the /etc/exports file but you want to export it differently.
	options specify the access restrictions. They can be `access`, `ro`, `rw`, and `anon`.
	path_name is the path name to the resource you want to export.
	Examples: The following `exportfs` command exports the /vol/vol1 volume to a netgroup named trusted_hosts, giving host1 root access and host2 read-write nonroot access:
	`exportfs -i -o access=trusted_hosts,root=host1,rw=host2 /vol/vol1`
	The following `exportfs` command exports the /vol/vol2 volume as a read-only volume to a netgroup named untrusted_hosts:
	`exportfs -i -o access=untrusted_hosts,ro /vol/vol2`

Unexporting resources

To unexport resources, complete the following steps.

Step	Action	
1	**If...**	**Then...**
	You want to unexport all resources that are currently exported	Go to Step 2.
	You want to unexport a specific resource that is currently exported	Go to Step 3.
2	Enter the following command: **`exportfs -au`**	

Step	Action
3	Enter the following command: **`exportfs -u path_name`** *path_name* is the path name to the resource that you want to unexport.

Providing secure NFS access

What secure NFS access does

Secure NFS access uses an authentication protocol to ensure the security of data and identity of users within a controlled domain.

Authentication protocol used

Data ONTAP provides secure NFS access using the Kerberos v5 authentication protocol.

Data ONTAP Kerberos KDC options

The Data ONTAP Kerberos v5 implementation supports the following Kerberos Key Distribution Center (KDC) options:

- A Windows 2000-based KDC, that is, a Windows 2000 network using Kerberos authentication. The Windows 2000 domain is the Kerberos realm.
- A UNIX-based KDC, that is, a UNIX network using Kerberos authentication. The UNIX domain is not always equivalent to the Kerberos realm.

Clients supported in a UNIX-based KDC

Kerberos v5 authentication supports the following clients:

- Solaris 2.6 running NFS version 2 (NFSv2) or NFS version 3 (NFSv3), with Solaris Easy Access Server (SEAM) 1.0
- Solaris 7 running NFSv2 or NFSv3, with Solaris Easy Access Server (SEAM) 1.0
- Solaris 8 running NFSv2 or NFSv3, with SEAM 1.01 (Solaris 8 Admin Pack) or the Solaris 8 Encryption Pack
- Solaris 9
- Windows clients running NFSv2 or NFSv3, with Hummingbird Maestro version 7
- Windows clients running NFSv2, NFSv3, or NFSv4, with Hummingbird Maestro version 8

Setting up security services using a Windows 2000-based KDC

The security service setup procedure adds a filer to a Windows 2000-based KDC as a service principal called nfs/*hostname.domain*@REALM.

Note

After you set up Windows 2000 security for NFS, Network Appliance does not support changing to an MIT KDC for NFS security.

To set up security service in a Windows 2000 network using the Kerberos protocol, follow these steps.

Step	Action
1	Edit the resource name in the /etc/exports file to include the security service you want. For information about exporting resources with security services, see “Specifying security services” on page 37. **Example:** `vol/vol0 -access=host1,sec=krb5`
2	Enter the following command: **`exportfs path_name`** *path_name* is a file or directory. **Example:** `exportfs vol/vol0`

Step	Action
3	Enter the following command: **`nfs setup`** **Result if CIFS is already set up:** The nfs setup program leads you through the following prompt: 1. `Enable Kerberos for NFS [y]:` The nfs setup program uses information you entered when you set up CIFS to connect to a Windows 2000 domain controller and completes the setup by enabling Kerberos authentication for NFS. **Result if CIFS is not set up:** The nfs setup program leads you through the first prompt followed by these prompts: 2. `Do you want to modify this name? [no]:` The nfs setup program uses the default filer name if you press Enter. You can change the name of the filer by entering the filer name at this prompt. 3. `Enter the Windows Domain for the filer []:` The setup program uses the Windows domain you enter as the Kerberos realm. 4. `Please enter the Windows 2000 user []:` The setup program requires an administrator account that can add the filer to the Windows domain. 5. `Password for:` The setup program requires the password for the administrator account. After you enter the password, the filer connects to the Windows 2000 domain controller and completes the setup by enabling Kerberos authentication for NFS.

Setting up security services using a UNIX-based KDC

The security service setup procedure adds a filer to a standard UNIX-based KDC as a service principal called nfs/*hostname.domain*@REALM.

Before setting up security services, make sure the following requirements are met:

- An NFS client and a UNIX-based KDC are set up, with client principals for root and at least one non-root client.
- NFS access is verified between the client and a non-NetApp server.

This procedure assumes that you are using MIT KDC software for creating the server principal and keytab on the NFS server.

Note

After you set up security for NFS using an MIT KDC, Network Appliance does not support changing to Windows 2000 security for NFS.

To set up security service in a UNIX-based network using the Kerberos protocol, follow these steps.

Step	Action
1	On the NFS server, at the kadmin or kadmin.local command prompt, enter the following command: **`ank -randkey nfs/`*`hostname.domain`*** *hostname* is the host name of the NFS server principal. *domain* is the domain of the NFS server principal. **Example:** `kadmin: ank -randkey nfs/server.lab.netapp.com` **Result:** A principal is created for the NFS server, for example, nfs/server.lab.netapp.com@LAB.NETAPP.COM, where the realm is @LAB.NETAPP.COM.

Step	Action
2	On the NFS server, from kadmin or kadmin.local, enter the following command: **`xst -k/tmp/filer.krb5.keytab nfs/`*`hostname.domain`*** *hostname* is the host name of the server principal created in Step 1. *domain* is the domain of the server principal created in Step 1. **Example:** `kadmin: xst -k/tmp/filer.krb5.keytab nfs/server.lab.netapp.com` **Result:** In this example, a keytab is created for the server principal nfs/server.lab.netapp.com@LAB.NETAPP.COM. The KVNO 3 encryption type DES-CBC-CRC is added to the keytab WRFILE:/tmp/filer.krb5.keytab.
3	On the NFS server, enter the following command: **`cp /tmp/filer.krb5.keytab /net/filer/vol/vol0/etc/krb5.keytab`** **Result:** The keytab is copied to the filer. **Note** Once the keytab is copied to the filer, be sure you do not NFS-export the /etc subdirectory of the volume. If you NFS-export the /etc subdirectory, NFS clients can read the key information and could masquerade as the filer.
4	To copy the krb5.conf file to the filer, do one of the following: ◆ On a UNIX client running MIT KDC software, enter the following command: **`cp /etc/krb5.conf /net/filer/vol/vol0/etc/krb5.conf`** ◆ On a Solaris client running SEAM, enter the following command: **`cp /etc/krb5/krb5.conf /net/filer/vol/vol0/etc/krb5.conf`**

Step	Action
5	On the filer, enter the following command: **`options nfs.kerberos.file_keytab.enable on`** **Result:** Enables use of a file-based keytab in /etc/krb5.keytab.
6	Enter the following command: **`options nfs.kerberos.principal hostname.domain`** *hostname* is the host name of the server principal created in Step 1. *domain* is the domain of the server principal created in Step 1. **Example:** `options nfs.kerberos.principal server.lab.netapp.com` **Note** Be sure that the host name and domain you enter is the same as the host name and domain you entered for the server principal in Step 1 and the keytab in Step 2. In this step, you do not enter "nfs/" before the host name.
7	Enter the following command: **`options nfs.kerberos.realm realm`** *realm* is the realm-specific part of the NFS server's Kerberos principal name. **Example:** `options nfs.kerberos.realm @NETAPP.COM`
8	Enter the following command: **`exportfs -i -o sec=krb5 path_name`** *path_name* is a file or directory. **Example:** `exportfs -i -o sec=krb5 /vol/vol0/home`

Enabling and disabling NFS versions

Enabling and disabling NFS versions

To enable or disable a specific version of NFS on your filer, complete the following step.

Step	Action
1	Enter the following command: **`options nfs.v`*`N`*`.enable on	off`** *N* is the number of the version you want to enable or disable. Value values for *N* are 2, 3, or 4. **Example:** `options nfs.v4.enable on` **Note** This option is turned on by default for NFSv2 and NFSv3 and turned off by default for NFSv4.

Using NFSv4

Support for NFSv4

Data ONTAP supports all of the mandatory functionality in NFSv4, except the SPKM3 and LIPKEY security mechanisms, including the following:

- COMPOUND—Allows a client to request multiple file operations in a single remote procedure call (RPC) request.
- Pseudo-fs—Used by NFSv4 servers to determine mount points on the filer. There is no mount protocol in NFSv4.
- Locking—Lease-based. There are no separate Network Lock Manager (NLM) or Network Status Monitor (NSM) protocols in NFSv4.
- Named attributes—Similar to Windows NT streams.

Limitations for NFSv4

Be aware of the following limitations when using NFSv4 with Data ONTAP:

- Delegations, which allow the NFS server to delegate the handling of files to the client, are not supported.
- Named attributes can only be created if CIFS is running on the filer.
- Names with non-ASCII characters on volumes other than UTF8 volumes are rejected by the filer.
- All file handles are persistent. The server does not give out volatile file handles.
- Migration and replication are not supported.
- NFSv4 access control lists (ACLs) are not supported. ACLs set by CIFS apply to NFSv4 access, as they do for other versions of NFS.
- All recommended attributes are supported, except for the following:
  - acl
  - aclsupport
  - archive
  - hidden
  - homogeneous
  - mimetype
  - quota_avail_hard
  - quota_avail_soft
  - quota_used
  - system

- time_backup

- NFSv4 does not use the User Datagram Protocol (UDP) transport protocol. If you enable NFSv4 and disable NFS over TCP by setting `options nfs.tcp.enable` to Off, then NFSv4 is effectively disabled.

How the pseudo-fs in NFSv4 affects mount points

NFSv4 uses a pseudo-fs as an entry point into the filer for determining mount points. A pseudo-fs allows you to use one port for security, rather than several. All NFSv4 servers support the use of a pseudo-fs.

You may experience inconsistencies with mount points between NFSv3 and NFSv4, because of the pseudo-fs used in NFSv4.

In the examples that follow, you have these volumes:

/vol/vol0 (root)
/vol/vol1
/vol/home

Example 1:

In NFSv3 if you do not use the complete path from /vol/vol0, and you mount filer:/, the mount point is filer:/vol/vol0. That is, if the path does not begin with /vol in NFSv3, then Data ONTAP adds /vol/vol0 to the beginning of the path.

In NFSv4, if you do not use the complete path from /vol/vol0 and you mount filer:/, you mount the root of the pseudo-fs and not /vol/vol0. Data ONTAP does not add /vol/vol0 to the beginning of the path.

Therefore, if you mount filer:/ /n/filer using NFSv3 and try the same mount using NFSv4, you would mount a different file system.

Example 2:

In Data ONTAP's implementation of the NFSv4 pseudo-fs, the nodes "/" and "/vol" are always present and form the common prefix of any reference into the pseudo-fs. Any reference that does not begin with "/vol" is invalid.

In this example, there is a /vol/vol0/home directory. In NFSv3, if you mount filer:/home /users, /home is considered as the directory /vol/vol0/home. In NFSv4, if you mount filer:/home /users, /home is not considered as the volume /vol/home; it is considered an invalid path in the pseudo-fs tree.

Specifying the user ID domain for NFSv4

The domain that Data ONTAP uses for NFSv4 user ID mapping by default is the NIS domain, if one is set. If a NIS domain is not set, the DNS domain is used. You might need to set the user ID domain if, for example, you have multiple user ID domains.

To specify the user ID domain to be used for NFSv4 user ID mapping, complete the following step.

Step	Action
1	Enter the following command: **`option nfs.v4.id_domain`** ***`domain`***

File sharing with (PC)NFS clients

What Data ONTAP supports for (PC)NFS clients

Data ONTAP supports the `pcnfsd` daemon, which provides authentication services for clients using (PC)NFS version 1 or 2. Authenticated (PC)NFS users can mount resources from the filer in the same way as NFS users. The `pcnfsd` daemon does not support printer service.

How pcnfsd authenticates users

When the `pcnfsd` daemon receives an authentication request, it can use local files or NIS maps to validate the user's password. The local file used can be the /etc/shadow file or /etc/passwd file. The NIS maps used can be passwd.adjunct or passwd.byname. Data ONTAP uses the shadow source, which contains encrypted user information, instead of the password database if the shadow source is available.

The following list describes how the `pcnfsd` daemon uses local files or NIS maps for authenticating both (PC)NFS version 1 and version 2 users:

- If a shadow source is available, Data ONTAP uses the /etc/shadow file or the passwd.adjunct NIS map to determine the user's password.
- If a shadow source is not available, Data ONTAP uses the /etc/passwd file or the passwd.byname NIS map to determine the user's UID, primary GID, and password.

How pcnfsd determines group membership

When the `pcnfsd` daemon receives a (PC)NFS version 2 authentication request, it looks up the /etc/group file or the group.byname NIS map to determine all the groups to which the user belongs.

For detailed information

The following sections describe how to configure Data ONTAP for delivering (PC)NFS service:

- "Enabling or disabling the pcnfsd daemon" on page 58
- "Creating (PC)NFS user entries on the filer" on page 59

Enabling or disabling the pcnfsd daemon

Purpose of enabling pcnfsd

Enable the `pcnfsd` daemon if you want the filer to authenticate (PC)NFS users when they try to mount resources from the filer. If you want another computer to authenticate users, you do not need to enable the `pcnfsd` daemon. Users authenticated by other computers can access resources on the filer in the same way as users authenticated by the filer.

Prerequisite for enabling the pcnfsd daemon

NFS must be enabled on the filer before you can enable the `pcnfsd` daemon.

Enabling or disabling pcnfsd

To enable or disable `pcnfsd`, complete the following step.

Step	Action
1	Enter the following command to enable or disable `pcnfsd`: **`options pcnfsd.enable on | off`** Specify `on` to enable `pcnfsd` or `off` to disable `pcnfsd`.

Creating (PC)NFS user entries on the filer

Reason for creating (PC)NFS user entries

If you want to use local files to authenticate users and to determine group membership, create (PC)NFS user entries in the local files. The local files used by the `pcnfsd` daemon are the /etc/passwd file, the /etc/shadow file, and the /etc/group file.

Creating (PC)NFS user entries

To create (PC)NFS user entries on the filer, complete the following steps.

Step	Action
1	Select a UNIX host with the /etc/passwd, /etc/shadow, and /etc/group files that contain all (PC)NFS users to be authenticated by the filer.
2	Copy the files from the UNIX host to the filer.

Defining the umask for (PC)NFS-created files and directories

About file permissions

Unlike NFS users, (PC)NFS users cannot execute the UNIX `umask` command to set the file mode creation mask (umask), which determines the default file permissions. However, Data ONTAP enables you to define the umask for all (PC)NFS users.

How umask works for (PC)NFS-created files

The permissions for each file are defined by three octal values, which apply to owner, group, and other. When a (PC)NFS client creates a new file, Data ONTAP subtracts the umask, which is a three-digit octal number you define, from 666. The resulting octal digits are used as file permissions.

By default, the umask is 022, which means that the effective octal digits for permissions are 644. These permissions enable the file owner to read and write the file, and enable the group and other only to read the file.

Meaning of each digit in the umask

The following table provides the description for each digit in the umask.

Digit in the umask	Description
0	Read and write permissions
2	Write permission
4	Read-only permission
6	No permission

Defining the umask

To define the umask, complete the following step.

Step	Action
1	Enter the following command: **`options pcnfsd.umask`** ***`umask`*** *umask* is a three-digit octal number.

Configuring Data ONTAP for WebNFS

About WebNFS

The filer can transfer files to clients using the WebNFS protocol. That is, in a Web browser that supports WebNFS, a user can type a URL starting with nfs:// to transfer a file from the filer.

Requirements for file transfers through WebNFS

For a file to be transferred using WebNFS, these requirements must be met:

- The file must be exported.
- The file must not be in a directory that is exported with the `access` option.

Root directory for WebNFS lookup

If you define a root directory for WebNFS lookup, a WebNFS user can type only the path name relative to the root directory instead of the absolute path starting with /. For example, if the WebNFS root directory is /vol/vol1/web, you can access the /vol/vol1/web/specs file by typing nfs://specs as the URL.

Configuring Data ONTAP for WebNFS

To configure Data ONTAP for WebNFS, complete the following steps.

Step	Action	
1	Enter the following command to enable or disable WebNFS: **`options nfs.webnfs.enable on	off`** Specify `on` to enable WebNFS or `off` to disable WebNFS.	
2	**If...**	**Then...**
	You disable WebNFS in Step 1	You are done.
	You enable WebNFS in Step 1	Go to Step 3.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="3">3</td><td>If...</td><td>Then...</td></tr>
<tr><td>You want to use relative path names in the URL</td><td>Go to Step 4.</td></tr>
<tr><td>You want to use complete path names in the URL</td><td>You are done.</td></tr>
<tr><td>4</td><td colspan="2">Enter the following command to specify a WebNFS root directory:
<tt>options nfs.webnfs.rootdir directory</tt>
directory is the complete path name of the WebNFS root directory.</td></tr>
<tr><td>5</td><td colspan="2">Enter the following command to make the WebNFS root directory go into effect:
<tt>options nfs.webnfs.rootdir.set true</tt></td></tr>
</table>

File Access Using CIFS 3

About this chapter

This chapter describes how to prepare your filer for delivering file service to CIFS users.

Topics in this chapter

This chapter discusses the following topics:

- "Filer management through Server Manager and User Manager" on page 65
- "Displaying and changing the description of the filer" on page 66
- "Creating NetBIOS aliases for the filer" on page 68
- "How to specify a Windows name" on page 70
- "How CIFS users obtain UNIX credentials" on page 72
- "Sharing directories on the filer" on page 95
- "Displaying and changing share properties" on page 102
- "Deleting a share" on page 104
- "CIFS limitations" on page 105
- "Understanding local user accounts" on page 108
- "Creating local groups on the filer" on page 110
- "Understanding home directories on the filer" on page 115
- "How Data ONTAP matches a directory with a user" on page 117
- "Setting up user home directories" on page 119
- "Accessing home directories" on page 127
- "Specifying support for wide symbolic links in CIFS home directories" on page 130
- "How to stop offering home shares" on page 131
- "How share-level access control lists work" on page 132
- "Specifying how GIDs work with share-level ACLs" on page 133
- "Changing and displaying a share-level ACL" on page 135
- "Changing and displaying file-level ACLs" on page 141
- "Understanding the SID-to-name map cache" on page 144
- "Understanding file screening" on page 150
- "Enabling and disabling file screening" on page 151
- "Managing file policies" on page 153
- "Managing file screening servers" on page 157
- "Sending a message to all users on a filer" on page 158

- "Understanding CIFS auditing" on page 161
- "Understanding oplocks" on page 177
- "Managing oplocks" on page 179
- "Displaying CIFS session information" on page 181
- "Timing out idle sessions" on page 183
- "Disabling CIFS service" on page 184
- "Restarting CIFS service" on page 190
- "Reconfiguring CIFS for the filer" on page 191
- "Selecting domain controllers" on page 195
- "Disabling NetBIOS over TCP" on page 198

Filer management through Server Manager and User Manager

Scope of Server Manager and User Manager descriptions

You can accomplish some tasks for managing CIFS file access from the Data ONTAP command line or from the Windows NT Server Manager or User Manager. This chapter describes both methods of accomplishing these tasks. This chapter assumes that you know how to start Server Manager or User Manager to access your filer.

Note

You can also accomplish the tasks described in this chapter using other Windows server management tools, such as Microsoft Management Console.

How to use Server Manager and User Manager to manage the filer

The procedures for managing the filer through Server Manager and User Manager are similar to those for managing a Windows server. If you are familiar with Server Manager and User Manager, you might not need to read the procedures involving them. The procedures in this chapter give detailed steps where there are differences between managing a filer and managing a Windows server.

Case-sensitivity in Server Manager and User Manager

Unlike text you type at the Data ONTAP command line, text you type in Server Manager and User Manager is not case-sensitive. For example, when you specify a volume name in Server Manager, you can type in either lowercase or uppercase letters.

Displaying and changing the description of the filer

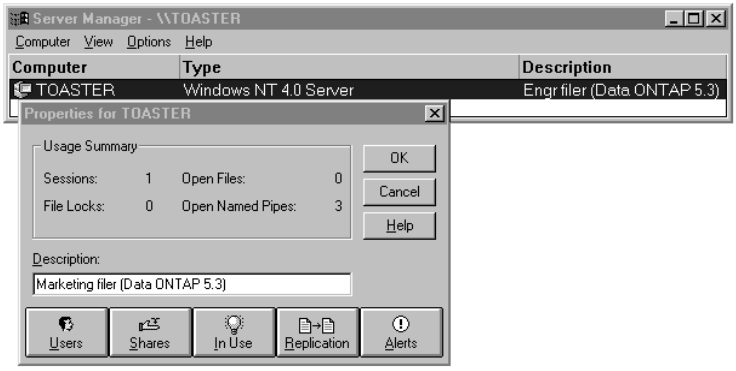
About the filer description

The description of a filer appears in the Comment field when you browse the network. Initially, the filer has no description. Adding an informative description enables you to distinguish your filer from other computers on the network.

The description can be up to 48 characters.

Changing the filer description from Server Manager

To change the filer description from Server Manager, complete the following steps.

Step	Action
1	Select your filer from Server Manager.
2	Type the description of the filer in the Description field. **Example:** The following figure illustrates how to change the description from "Engr filer (Data ONTAP 5.3)" to "Marketing filer (Data ONTAP 5.3)." 

Displaying and changing a filer description from the command line

To display and change the description of a filer from the command line, complete the following steps.

Step	Action
1	Enter the following command to display the current description: `cifs comment`
2	Enter the following command to change the description: `cifs comment "description"`

Creating NetBIOS aliases for the filer

About NetBIOS aliases

You can create a list of NetBIOS aliases as alternative names for the filer. You can connect to the filer using any of the names in the list.

Ways to create NetBIOS aliases

You can create NetBIOS aliases in either of the following ways:

- Using the command `options cifs.netbios_aliases`
- Using the /etc/cifs_nbalias.cfg file

The `options cifs.netbios_aliases` command allows you to create NetBIOS aliases as a comma-separated list. The list has a limit of 255 characters, including the commas. Alternatively, you can create up to 200 entries in the /etc/cifs_nbalias.cfg file.

Creating NetBIOS aliases from the command line

To create a list of NetBIOS aliases from the command line, complete the following step.

Step	Action
1	Enter the following command: **`options cifs.netbios_aliases`** ***`name,...`*** **Example:** `options cifs.netbios_aliases alias1,alias2,alias3` **Note** You can enter a list of aliases up to a limit of 255 characters, including the commas. If the aliases you want to create exceed 255 characters, you can create the aliases in a file named cifs_nbalias.cfg in the /etc directory.
2	Enter the following command to process the entries: **`cifs nbalias load`**

About the default cifs_nbalias.cfg file

Data ONTAP creates a default cifs_nbalias.cfg file in the /etc directory when CIFS starts, if the file does not already exist. Changes to this file are processed automatically whenever CIFS starts. You can also process changes to this file using the command `cifs nbalias load`.

Creating NetBIOS aliases in the cifs_nbalias.cfg file

To create NetBIOS aliases in the /etc/cifs_nbalias.cfg file, complete the following steps.

Step	Action
1	Open the /etc/cifs_nbalias.cfg file for editing.
2	Enter NetBIOS aliases in the /etc/cifs_nbalias.cfg file, one entry per line. **Note** You can enter up to 200 NetBIOS aliases in the file using either ASCII or Unicode characters.
3	Enter the following command to process the entries: **`cifs nbalias load`**

Displaying the list of NetBIOS aliases

To display the current list of NetBIOS aliases, complete the following step.

Step	Action
1	Enter the following command: **`cifs nbalias`**

How to specify a Windows name

Where Windows names are specified

You specify Windows names in some Data ONTAP commands and configuration files. The following examples show where you can specify Windows names:

- You can use a Windows name as the argument to the `cifs sessions` command to display information about a Windows user.
- You can specify Windows names in the /etc/usermap.cfg file to map Windows names to UNIX names.
- You can specify Windows names in the /etc/quotas file to establish quotas for Windows users.

Pre–Windows 2000 format

You can specify a Windows name in the pre–Windows 2000 format. In this format, the domain name is followed by a backslash and user name, for example, corp\john_smith.

Conversion from Windows 2000 format to pre–Windows 2000 format

You can specify the name of a Windows 2000 user in the pre–Windows 2000 format. When converting a Windows name to the pre–Windows 2000 format, remember these rules:

- The user name must not exceed 20 characters.
- The NetBIOS form of the domain name must be used.

Example: If john_smith@engineering.netapp.com is a Windows 2000 user, you can refer to this user as engineering\john_smith in Data ONTAP commands and configuration files.

Format for a local user account

You can set up local user accounts on the filer, as described in "Understanding local user accounts" on page 108. To specify a local user account, replace the domain name with the filer name in the pre–Windows 2000 format, for example, filer1\john_smith.

The \ character in user names

If you specify a UNIX user name with a backslash (\) in a configuration file, Data ONTAP treats the name as a Windows name. For example, UNIX names such as corp\john in the /etc/quotas file are interpreted as Windows names.

Rules for user names specific to certain commands or files

The only command in which you can specify user names using the *user@domain* format is the `cifs setup` command. There might be additional rules for specifying Windows names that are specific to particular configuration files. For additional information about those rules, see the sections relating to the particular configuration files.

How CIFS users obtain UNIX credentials

About UNIX credentials

When connecting to the filer, the user on the CIFS client receives a CIFS credential. The user must also have one or more UNIX credentials. A UNIX credential consists of a UNIX-style UID and group IDs (GIDs). When a CIFS user tries to connect to the filer, Data ONTAP determines the UID and primary GID of the CIFS user. If Data ONTAP cannot determine the UID of the CIFS user, the user is denied access.

You can see the UNIX credential of a connected CIFS user when you display CIFS session information, as described in "Displaying CIFS session information" on page 181.

How Data ONTAP uses the UNIX credential

Data ONTAP uses the UNIX credential in these ways:

- Data ONTAP uses the UID and GID to determine the access rights of the user when the user tries to access files with UNIX-style security.
- If you want to use group quotas on a group that contains CIFS users, the CIFS users must have UNIX credentials. For more information about group quotas, see the chapter about disk space management using quotas in the *Storage Management Guide*.

How Data ONTAP obtains the UNIX credential

Data ONTAP looks up the UNIX password database, which can be an NIS map or the /etc/passwd file, to obtain the UID for a user. The database contains accounts for all users that might access the filer. Each account contains a UNIX-style user name and UID.

For Data ONTAP to obtain a UID for a CIFS user, it must first determine the user's UNIX-style name. Data ONTAP does not require that a user's Windows name be identical to the UNIX name. By entering information in the /etc/usermap.cfg file, you can specify how each Windows name maps to a UNIX name. You do not, however, need to enter mapping information in the /etc/usermap.cfg file if you accept the default mapping. By default, Data ONTAP uses the Windows name as the UNIX name when it looks up the UID. (The filer converts uppercase characters in the Windows name to lowercase before the lookup.) If the user names in the UNIX password database are identical to the Windows names, you need not provide the mapping information in the

/etc/usermap.cfg file. If the user name is not found in the UNIX password database and the `wafl.default_unix_user pcuser` option is set, the default name pcuser is used.

How Data ONTAP obtains the GIDs

Data ONTAP obtains a user's GIDs in the following ways:

- Data ONTAP obtains the user's primary GID from the UNIX password database. Each account in the UNIX password database contains the GID for that user.
- Data ONTAP obtains the user's other GIDs from the group database, which can be the NIS group map or the /etc/group file. The group database is where you define membership for various groups.

Ensuring that only intended CIFS users receive UNIX credentials

To ensure that only the intended CIFS users receive UNIX credentials and can access the filer, complete the following steps.

Step	Action	
1	**If...**	**Then...**
	Some Windows names are different from UNIX names or you want to prevent some CIFS users from accessing the filer	Edit the /etc/usermap.cfg file, and then go to Step 2. See "How Data ONTAP maps names" on page 76.
	All Windows names are the same as UNIX names and you want all CIFS users to be able access the filer	Go to Step 2.
2	Create groups in the UNIX group database. See "Mapping UNIX names to UIDs and GIDs" on page 89.	
3	For each CIFS user with a mapped UNIX name, enter the user account in the UNIX password database.	

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="3">4</td><td>If...</td><td>Then...</td></tr>
<tr><td>You rename the Administrator account</td><td>Make sure at least one CIFS user maps to the UNIX root account, and then go to Step 5.

See “Mapping a Windows account to root” on page 87.

Note
If you set the option <code>wafl.nt_admin_priv_map_to_root to</code> On, all accounts in the Administrators group are considered root. You do not need to complete the following steps.</td></tr>
<tr><td>You do not rename the Administrator account</td><td>Go to Step 5.</td></tr>
<tr><td rowspan="3">5</td><td>If...</td><td>Then...</td></tr>
<tr><td>You want CIFS users who do not have an entry in the UNIX password database to access the filer</td><td>Create a default user account in the UNIX password database, and then go to Step 6.

See “Creating or disabling the default UNIX user account” on page 92.</td></tr>
<tr><td>You want all CIFS users who have an entry in the UNIX password database to access the filer</td><td>Go to Step 6.</td></tr>
</table>

Step	Action	
6	**If...**	**Then...**
	You want unauthenticated users to access the filer	See "Enabling or disabling the Windows guest user account" on page 94.
	You do not want unauthenticated users to access the filer	You are done.

How Data ONTAP maps names

File used for mapping names

Data ONTAP uses the /etc/usermap.cfg file to map names. In its simplest form, each /etc/usermap.cfg entry contains a pair of names: the Windows name and the UNIX name. Data ONTAP can translate the Windows name to the UNIX name or vice versa.

How Data ONTAP uses the /etc/usermap.cfg file

When Data ONTAP receives a connection request from a CIFS user, it searches the /etc/usermap.cfg file to see whether an entry matches the user's Windows domain name and user name.

If an entry is found: Data ONTAP uses the UNIX name specified in the entry to look up the UID and GID from the UNIX password database. If the UNIX name is a null string, Data ONTAP denies access to the CIFS user.

If an entry is not found: Data ONTAP converts the Windows name to lowercase and considers the UNIX name to be the same as the Windows name. Data ONTAP uses this UNIX name to look up the UID and GID from the UNIX password database.

Note

Data ONTAP scans the file sequentially. It uses the first matching entry for mapping.

How to specify entries for the /etc/usermap.cfg file

Character coding of the /etc/usermap.cfg file

For information about character coding of the /etc/usermap.cfg file, see the section about the contents of the /etc directory in the *Storage Management Guide*.

Overview of the /etc/usermap.cfg format

Each line in the /etc/usermap.cfg file is a map entry in the following format:

`[IP_qualifier:] Windows_name [direction] [IP_qualifier:]UNIX_name`

The direction field determines whether the entry is for mapping from Windows to UNIX, from UNIX to Windows, or both. For information about why Data ONTAP needs to map UNIX names to Windows names, see "Accessing CIFS files from NFS clients" on page 315.

You can embed comments in the file by beginning the comment lines with #. Comments at the end of an entry are also allowed if preceded by #. Blank lines are ignored.

Default contents of the /etc/usermap.cfg file

When CIFS is started, if the /etc/usermap.cfg file is missing, a default file is created. It contains commented-out sample map entries that are useful for improving security.

IP_qualifier field

The IP_qualifier field is an IP address that qualifies the user name. It narrows the match.

The IP qualifier can be

- An IP address in bit notation. You can specify a subnet by including the number of bits in the subnet mask. For example, 192.4.1.0/24 means the 192.4.1.0 class C subnet.
- A name. Data ONTAP first considers a name to be a host name. If it cannot find a matching host name in its host name database, it considers the name to be a network name.

- A subnet address that includes a network name or IP address and the subnet mask (for example, corpnet/255.255.255.0).

Note

Data ONTAP uses the IP qualifier only for matching. If an IP qualifier is present on the destination side of a map entry, Data ONTAP does not consider the login request to come from that IP qualifier.

Windows_name field

The Windows_name field consists of a Windows domain name, which is optional, and a Windows user name. For more information about the syntax, see "How to specify a Windows name" on page 70.

Meaning of Windows domain: On the source side of the map entry, the domain specifies the domain in which the user resides. On the destination side of the map entry, it specifies the domain used for the mapped UNIX entry. If the account name in the entry is a local user account, the Windows domain name is the filer name.

Default Windows domain: If you omit the domain name in the Windows_name field, it is assumed to be the domain in which the filer is installed. If the filer uses local user accounts for authentication, the domain name is the filer name.

Use of wildcard character in the domain name: You can use an asterisk (*) in the following ways:

- You can use it on the source side to indicate that the specified name in any domain maps to the specified UNIX name.
- You can use it on the destination side to indicate that the specified UNIX name maps to a Windows name in any trusted domain.

  The trusted domain used for the mapping depends on where Data ONTAP finds the first matching Windows name. Data ONTAP searches only the trusted domains you specify with the `cifs.search_domains` option, in the order in which the trusted domains are specified. If you do not set this option, Data ONTAP searches all trusted domains in an unspecified order.

Special characters in the user name: If the user name contains spaces or a pound sign (#), enclose the name in double quotation marks, for example, "bob smith" or "eng#lab"\"#joe".

Note

Only the user name should be inside the double-quotation marks. For example, it is incorrect to specify “mktg\bob smith”. The correct syntax is mktg\“bob smith”.

Use of wildcard character in the user name: You can use an asterisk (*) in the Windows name. For more information about how to use the asterisk, see “Guidelines for wildcard character in user name” on page 80.

Use of blank as the user name: If the user name is empty or blank (specified as “”) on the destination side of the map entry, the matching UNIX name is denied access. Use entries with a blank user name to deny access to some or all UNIX users. If you use these entries in conjunction with IP_qualifier, you can exclude all UNIX users except for certain hosts or subnets.

Direction field

The direction field indicates the direction of the mapping. It can be one of the values in the following table.

Value of the direction field	Meaning
==	Mapping is bidirectional. The entry maps from Windows to UNIX and from UNIX to Windows. Omitting the direction field has the same meaning as specifying ==.
<=	The entry maps from UNIX to Windows.
=>	The entry maps from Windows to UNIX.

UNIX_name field

Meaning of UNIX_name: The UNIX_name field is a UNIX name in the UNIX password database.

Use of wildcard character in the UNIX name: You can use an asterisk (*) in the UNIX name. For more information about how to use the asterisk, see “Guidelines for wildcard character in user name” on page 80.

Use of blank as the UNIX name: If UNIX_name is empty or blank (specified as “”) on the destination side of the map entry, the specified source name is prevented from logging in. The Windows user cannot log in to the filer even if the user can see the filer while browsing the network.

Guidelines for wildcard character in user name

The asterisk is considered the wildcard character. It means any user. Remember these guidelines when including an asterisk in the Windows name or the UNIX name:

- If the asterisk is on the source side of the mapping, any user maps to the specified name on the destination side.
- If the destination side contains an asterisk but the source side does not, no mapping is done. Data ONTAP does not map an explicitly specified name to a name with an asterisk.
- If both the source and destination sides contain an asterisk, the corresponding name is mapped.

How Data ONTAP interprets domain names in /etc/usermap.cfg

Factors affecting how domain names are interpreted

In the /etc/usermap.cfg file, there might be domain names that contain a dot. The following list describes how Data ONTAP interprets these domain names:

- If the filer is installed in a pre–Windows 2000 domain, the length of the domain name field affects how the domain name is interpreted.
- If the filer is installed in a Windows 2000 domain, Data ONTAP interprets the domain names in the same way as a Windows 2000 server does.

Pre–Windows 2000 domain

If the filer is in a pre–Windows 2000 domain, Data ONTAP follows these rules when interpreting a domain name containing a dot in the *domain\user* format:

- If *domain* is 15 characters or shorter, Data ONTAP recognizes the entire string, including the dot, as the NetBIOS form of the domain name. For example, netapp.com is the NetBIOS form of the domain name in the following name:

  netapp.com\john_smith

- If *domain* is longer than 15 characters, the dot is treated as a separator, and the string before the first dot is the NetBIOS form of the domain name. For example, engineering is the NetBIOS form of the domain name in the following name:

  engineering.1234567890corporation.com\john_smith

Windows 2000 domain

If the filer is in a Windows 2000 domain, you can specify a user name in the *domain\user* format. The string before the first dot in *domain* is the NetBIOS form of the domain name, and the entire string in *domain* is the DNS domain name.

For example, engineering is the NetBIOS form of the domain name and engineering.1234567890corporation.com is the DNS domain name in the following name:

engineering.1234567890corporation.com\john_smith

Examples of usermap.cfg entries

Examples of simple usermap.cfg entries

The following table describes some simple /etc/usermap.cfg entries.

Entry	Meaning
“Bob Garj” == bobg	The Windows name Bob Garj maps to the UNIX name bobg and vice versa.
mktg\Roy => nobody	The Windows name Roy in the mktg domain maps to the UNIX name nobody. This entry enables Roy to log in with limited access to files with UNIX-style security.
engr\Tom => “”	Disallow login by the user named Tom in the engr domain.

Examples with asterisks

The following table provides some examples with asterisks in the Windows names.

Entry	Meaning
uguest <= *	All UNIX names not yet matched map to Windows user uguest.
*\root => “”	Disallow logins using the Windows name root from all domains.
corporate* == pcuser	Any user in the corporate domain maps to the UNIX name pcuser. No mapping is done for the UNIX name pcuser because an asterisk is used in the Windows user name.
Engineer == *	Any UNIX name maps to the Windows name Engineer in the filer’s domain. No mapping is done for the Windows name Engineer because an asterisk is used in the UNIX user name.

Entry	Meaning
Either of the following entries: ◆ homeusers* * ◆ homeusers* == *	All UNIX users map to the corresponding names in the homeusers domain. For example, a UNIX user named bob maps to homeusers\bob. All Windows users from the homeusers domain map to their corresponding UNIX names. For example, a Windows user named john in the homeusers domain maps to the UNIX name john.

Examples with IP qualifiers

The following table provides some examples with IP qualifiers.

Entry	Meaning
Engineering* <= sunbox2:*	UNIX names from the host named sunbox2 map to the same names in the Engineering domain.
Engineering* <= 192.9.200.70:*	UNIX names from the IP address 192.9.200.70 map to the same names in the Engineering domain.
“”<= 192.9.200.0/24:*	All NFS requests from the 192.9.200.0 subnet are denied because UNIX names from this subnet map to a null string.
192.9.200.0/24:test-dom* => “”	All users in the test-dom domain are denied access from the 192.9.200.0 subnet.

Entry	Meaning
** == corpnet/255.255.0.0:*	All user names from all domains map to the corresponding UNIX names. If user names are not unique across domains, this entry might cause different Windows names to map to the same UNIX name. Because IP qualifiers are only for matching, specifying corpnet/255.255.0.0: does not affect the result of Windows to UNIX mapping. Because the mapping is bidirectional, all UNIX names from the corpnet/255.255.0.0 network map to the same names in one of the filer's trusted domains.

Guidelines and recommendations for mapping users

Guidelines for mapping user names

Follow these guidelines to keep entries simple and easy to understand:

- Keep Windows user names and UNIX user names the same whenever possible. If the names are identical, you do not need to create map entries in the /etc/usermap.cfg file.
- Avoid creating confusing map entries such as these:

  ```
  "tome s" => tjs
  ```

  ```
  bill <= tjs
  ```

- Avoid using IP qualifiers to map users differently. For example, it is confusing if you map UNIX user tjs from UHOST1 to Windows user "Tom S" but UNIX user tjs from UHOST2 to Windows user Smith. Use IP qualifiers only to restrict access.

Recommended entries for increased security

The entries in the following table help prevent unauthorized users from accessing the filer. Remember that the order of entries is important when you copy these recommended entries to your file, because Data ONTAP uses the first matching entry to determine the mapping.

Map entry	Meaning
*\root => nobody	Any Windows users named root can log in, but they do not have UNIX rights. For any instances of a Windows user named root that should map differently, you explicitly add a map entry earlier in the /etc/usermap.cfg file.

Map entry	Meaning
guest <= administrator guest <= root	The first entry prevents spoofing the Windows Administrator account from UNIX (if the Administrator account has not been renamed). The second entry maps the UNIX user root to the Windows guest account. Type the second entry near the end of the /etc/usermap.cfg file after any explicit map entries for root-privileged UNIX hosts or subnets.
** => “” “” <= *	These entries, placed at the end of the file, prevent any other mapping from occurring. They defeat the default behavior that if an entry is not matched, the same name is tried.

Mapping a Windows account to root

Importance of mapping a Windows account to a UNIX root account

If you have only CIFS clients in your environment and your filer was set up as a multiprotocol filer, you must have at least one Windows account that has root privilege for accessing files on the filer. Otherwise, you cannot manage the filer because you do not have access to files with UNIX-style security, which include some configuration files in the /etc directory.

If your filer was set up as NTFS-only, however, the /etc directory has a file-level ACL that enables the Administrators group to access the Data ONTAP configuration files.

Mapping a Windows account to root

To map at least one Windows account to root, complete the following steps.

Step	Action	
1	**If...**	**Then...**
	You want to map all accounts in the Administrators group to root	Complete only Step 2.
	You want selected accounts to map to root	Complete Step 3 and Step 4.
2	Verify that the `wafl.nt_admin_priv_map_to_root` option is set to On. **Result:** All accounts in the Administrators group are considered root, even if you do not have an /etc/usermap.cfg entry mapping the accounts to root. If you create a file using an account that belongs to the Administrators group, the file is owned by root when you view the file from a UNIX client.	

Step	Action
3	For each account that maps to root, add an /etc/usermap.cfg entry. **Note** It is important to have at least one Windows account that maps to root on a multiprotocol filer. Otherwise, no accounts can access the configuration files in the /etc directory.
4	Disable the `wafl.nt_admin_priv_map_to_root` option with the following command: **`options wafl.nt_admin_priv_map_to_root off`** **Result:** Accounts in the Administrators group no longer map to root. You can use only those accounts that you map to root in the /etc/usermap.cfg file to access files with UNIX-style security. Each account in the Administrators group has a separate UNIX ID.

Mapping UNIX names to UIDs and GIDs

Where Data ONTAP obtains UIDs and GIDs

For each UNIX name, Data ONTAP obtains the UID and the primary GID from the UNIX password database. Data ONTAP obtains secondary GIDs for the UNIX name from the UNIX group database.

For a CIFS user to have a UID and GIDs, you must create a UNIX account in the UNIX password database that corresponds to the user's UNIX name.

Users not in the password database

A CIFS user whose UNIX name does not exist in the password database can still obtain a UID if certain requirements are met. For information about CIFS users whose UNIX names are not in the password database, see "Creating or disabling the default UNIX user account" on page 92.

When a default /etc/passwd file is created

If your filer is an NIS client before you run `cifs setup`, Data ONTAP does not automatically create the /etc/passwd file. If NIS is not enabled when you run `cifs setup`, Data ONTAP automatically creates the /etc/passwd file.

Note

If the NIS server fails and the filer does not have the /etc/passwd file, CIFS users cannot connect to the filer. You can create the /etc/passwd file to ensure that the filer can obtain UNIX credentials for CIFS users even when NIS is unavailable.

Contents of the default /etc/passwd file

The default /etc/passwd file contains entries for these UNIX names:

- root
- pcuser
- nobody

Format of the /etc/group and /etc/passwd files

For information about the format of the /etc/group and /etc/passwd files, see the *Network Management Guide*.

Creating UNIX accounts for CIFS users with UNIX names

To create UNIX accounts for CIFS users, complete the following steps.

Step	Action	
1	**If you use...**	**Then...**
	NIS but not the /etc/passwd file	Go to Step 2.
	The /etc/passwd file but not NIS	Go to Step 3.
2	Add the UNIX name of each CIFS user to the NIS password map. You are done.	
3	Add an entry in the /etc/passwd file for the UNIX name of each user. Because Data ONTAP does not support a command for creating a password entry, use one of these methods to create the /etc/passwd file: ◆ Use a UNIX host that supports the `passwd` command to create the /etc/passwd file on the host. Then copy the file from the host to the filer. ◆ Use SecureShare® Account Migrator to convert users and groups from a Windows domain into the format necessary to populate the /etc/passwd and /etc/group files. See “Migrating CIFS user accounts to the passwd and group files” on page 91.	

Migrating CIFS user accounts to the passwd and group files

About SecureShare Account Migrator

SecureShare Account Migrator is a supported tool for copying CIFS user account information from a domain controller to the /etc/passwd and /etc/group files on the filer. By copying the information, you can avoid manually entering account information in the files.

Locations of SecureShare Account Migrator

You can obtain SecureShare Account Migrator from these locations:

- The NOW site (http://now.netapp.com), where the installation procedure is also available
- The Data ONTAP CD-ROM

Running SecureShare Account Migrator from the CD-ROM

To run SecureShare Account Migrator from the CD-ROM, complete the following steps.

Step	Action
1	Insert the Data ONTAP CD-ROM in the CD-ROM drive of a PC that can access the domain controller.
2	Click the CD-ROM drive icon.
3	Open the Utils\Ssacmigr folder.
4	Double-click Ssamwin.exe.

Creating or disabling the default UNIX user account

Reason for creating a default UNIX user account

Create a default UNIX user account if there are users who need to connect to the filer occasionally but do not need to have individual entries in the UNIX password database. These users can use the default user account to connect to the filer.

How the default user account works with quotas

If quotas are enabled, the default user account is subject to quota restrictions in the same way as other users. For example, if the default user name is pcuser and a default user quota applies to the /vol/vol0 volume, pcuser is restricted by this default user quota. For more information about quotas for the default user, see the section about how disk space owned by default users is counted in the chapter about disk space management using quotas in the *Storage Management Guide*.

Prerequisites for accessing the filer as a default user

For a user to connect to the filer using the default user account, the user must meet the following prerequisites:

- The user is authenticated.
- The user is in a trusted domain.
- The user name does not map to a null string in the /etc/usermap.cfg file.

Default UNIX user name

The default UNIX name of the default user is pcuser. You can specify another name through the `wafl.default_unix_user` option. If this option is set to a null string, no one can access the filer as a UNIX default user. That is, each user must have an account in the password database before they can access the filer.

Creating or disabling a default user account

To create or disable a default user account, complete the following steps.

Step	Action	
1	**If...**	**Then...**
	You do not want any user to connect to the filer as a default user	Go to Step 2.
	You accept the default name for the default user account, which is pcuser	Go to Step 4.
	You want to use a name other than pcuser for the default account	Go to Step 3 and then Step 4.
2	Enter the following command: `options wafl.default_unix_user ""` You are done.	
3	Set the `wafl.default_unix_user` option to another name. **Example:** The following command changes the default user name to someuser: `options wafl.default_unix_user someuser`	
4	Create a default user account in the UNIX password database. For example, if you use pcuser as the default user account name, create an entry either in the NIS password database or the /etc/passwd file for pcuser.	

Enabling or disabling the Windows guest user account

Effect of enabling the guest user account

The effect of enabling the Windows guest user account depends on how the filer authenticates users, as explained in the following list:

- If the filer uses the domain controller or local user accounts to authenticate users, enabling the Windows guest user account means that users who log in from untrusted domains can connect to the filer. These users use the UNIX UID that you create specifically for the Guest account. A user logged in as Guest does not have a home share.
- If the filer uses the UNIX password database to authenticate users, enabling the Windows guest user account has the same effect as enabling the default UNIX account, except that the user logged in as Guest does not have a home share. For more information about the default UNIX account, see "Creating or disabling the default UNIX user account" on page 92.

Creating or disabling a guest user account

To create or disable a guest user account, complete the following steps.

Step	Action	
1	**If...**	**Then...**
	You do not want any user who is Guest to connect to the filer	Complete only Step 2.
	You want to use a guest account	Complete Step 3 and Step 4.
2	Enter the following command: `options cifs.guest_account ""`	
3	Enter the following command to specify the guest user account name used in the UNIX password database: `options cifs.guest_account unix_name` *unix_name* is the name of the user account in the UNIX password database.	
4	Create a user account in the NIS password database or the /etc/passwd file to be used by the guest account.	

Sharing directories on the filer

About shared directories

As Administrator, you can create directories on the filer. These directories do not automatically become accessible to users. You must create shares that correspond to these directories so that users can share them.

What you must specify when creating a share

You must provide

- The complete path name of an existing folder or qtree to be shared
- The name of the share used by users when they connect to the share

Share naming conventions

Share naming conventions for Data ONTAP are the same as for Windows. For example, share names ending with the $ character are hidden shares, and certain share names, such as ADMIN$ and IPC$, are reserved.

Share names are not case-sensitive.

What you can specify when creating a share

Optionally, you specify these share properties when you create a share:

- Description of the share.
- Maximum number of users who can simultaneously access the share.
- Share-level ACL. You can specify the share-level ACL during share creation only if you create the share from Server Manager. You can modify the share-level ACL of an existing share from Server Manager or from the filer's command line.

If you create the share from the Data ONTAP command line, you can additionally specify the following share properties:

- Group membership for files in the share.
- Control of share boundary checking for symbolic links in the share.
- Support for wide symbolic links in the share.
- Umask value for the share.
- Disable virus scanning when files in the share are opened.
- Disable virus scanning when files in the share are opened for read-only access.

You can change these attributes at any time after you create a share.

About the share description

The share description appears in the Comment field when you browse the shares on the network. The description can contain only characters that are in the current code page.

About the maximum number of users

You can specify the maximum number of users who can simultaneously access the share. If you do not specify the number, the number of users is the maximum allowed by the filer memory. For more information about how filer memory affects the number of users who can connect to the filer simultaneously, see "CIFS limitations" on page 105.

About group membership of files in the share

When you create a share from the Data ONTAP command line, you can specify that all files created by CIFS users in that share belong to the same group, which must be a predefined group in the UNIX group database. The group you specify is called a "forcegroup."

Effects of a forcegroup on CIFS access: The following list describes what happens to files created by CIFS if the share has a forcegroup:

- All files in this share created by CIFS users belong to the same forcegroup, regardless of the GID of the file owner.
- CIFS users who access this share are temporarily changed to the GID of the forcegroup. This GID enables them to access files in this share that are not accessible normally with their primary GID.

When CIFS users try to access a file created by NFS, the CIFS users' primary GIDs determine access rights.

Effects of a forcegroup on NFS access: The forcegroup does not affect how NFS users access files in this share. A file created by NFS acquires the GID from the file creator.

Advantages of specifying a forcegroup: Using a forcegroup makes it easier to ensure that files can be accessed by CIFS users belonging to various groups. For example, if you want to create a share to store the company's Web pages that can be written by users in Engineering and Marketing, create a share with a forcegroup named "web." Because of the forcegroup, all files created by CIFS users in this share are owned by the web group. In addition, users are automatically assigned the GID of the web group when accessing the share. As a result, they can all write to this share without your managing the access rights of the Engineering and Marketing groups.

Where a forcegroup is useful: Specifying a forcegroup is meaningful only if the share is in a UNIX or mixed qtree. There is no need to use forcegroups for shares in an NTFS qtree because access to files in these shares is determined by Windows permissions, not GIDs.

About the share-level ACL

When you create a share using Server Manager, you can specify the share-level ACL. If you use the Data ONTAP command line, you can specify the share-level permissions only after a share has been created. By default, a newly created share gives everyone Full Control. For more information about share-level ACLs, see "How share-level access control lists work" on page 132.

About share boundary checking for symbolic links

You disable share boundary checking for symbolic links for a share if you want to allow CIFS clients to follow symbolic links in that share to destinations anywhere on the same filer. If share boundary checking is disabled, the filer only checks the share permissions of the share that has the symbolic link. By default, the filer does share boundary checking for symbolic links to prevent them from leaving the share.

When you create a share from the Data ONTAP command line, you can disable share boundary checking for symbolic links by specifying `-nosymlink_strict_security`. When you change share properties from the command line, you can specify `-symlink_strict_security` or `-nosymlink_strict_security` to enable or disable share boundary checking for symbolic links.

For details about disabling share boundary checking for symbolic links, see "About disabling share boundary checking for symbolic links" on page 306.

About wide symbolic links

You enable wide symbolic links for a share if you want to allow CIFS clients to follow absolute symbolic links to destinations outside the share or outside the filer. By default, this feature is disabled.

When you create a share from the Data ONTAP command line, you can add support for wide symbolic links by specifying `-widelink`. When you change share properties from the command line, you can specify either `-widelink` or `-nowidelink` to enable or disable wide symbolic links.

For more information about how to create Widelink entries and additional requirements, see "About Widelink entries" on page 304. For more information about how to enable support for wide symbolic links in home directory shares, see "Specifying support for wide symbolic links in CIFS home directories" on page 130.

About the umask value

You can set the file mode creation umask for shares in qtrees with either UNIX or mixed security styles. The umask determines the permissions setting of newly created files and must be an octal value. The default umask value is 0.

When you create a share from the Data ONTAP command line, you can specify the umask using the command `-umask` *`mask`*. When you change share properties from the command line, you can reset the umask value back to 0 using the command `-noumask` or specify a different umask value using `-umask` *`mask`*.

Example: The following example turns off write access to "group" and "other" permissions when a file is created.

```
-umask 022
```

Note

The CIFS share umask value does not affect NFS.

About virus scanning

You can specify whether the filer performs a virus scan when clients open files on a share. You can also specify whether a virus scan is performed when clients open files for read-only access. Virus scanning for all opened files is turned on by default for new shares.

When you create a share from the Data ONTAP command line, you can turn virus scanning off as follows:

- For all files opened on the share by specifying `-novscan`
- Only for files opened for read-only access by specifying `-novscanread`

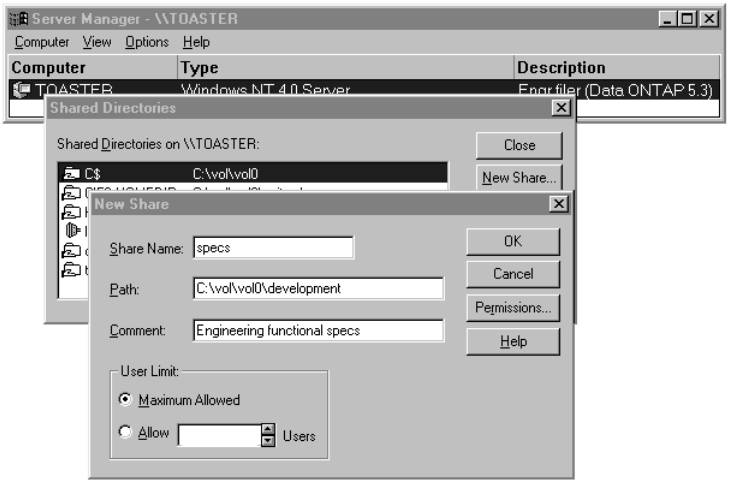
When you change share properties from the command line, you can turn virus scanning on or off as follows:

- For all files by specifying `-vscan` or `-novscan`
- Only for files opened for read-only access by specifying `-vscanread` or `-novscanread`

For more information about specifying virus scanning for CIFS shares, see the *Data Protection Guide*.

Creating a share from Server Manager

To create a share using Server Manager, complete the following steps.

Step	Action
1	Create a folder or a qtree on the filer.
2	Select your filer from Server Manager.
3	Create a share as you would on a Windows NT server. You can specify the maximum number of users and share-level ACLs for the share. **Note** In the Path field of the New Share window, be sure to type the complete path name of the folder or qtree you created in Step 1, starting with the C:\vol\vol_name prefix. C: is ignored by the filer. **Example:** The following figure illustrates how to create a new share named specs that corresponds to the /vol/vol0/development directory on the filer. 

Creating a share from the Data ONTAP command line

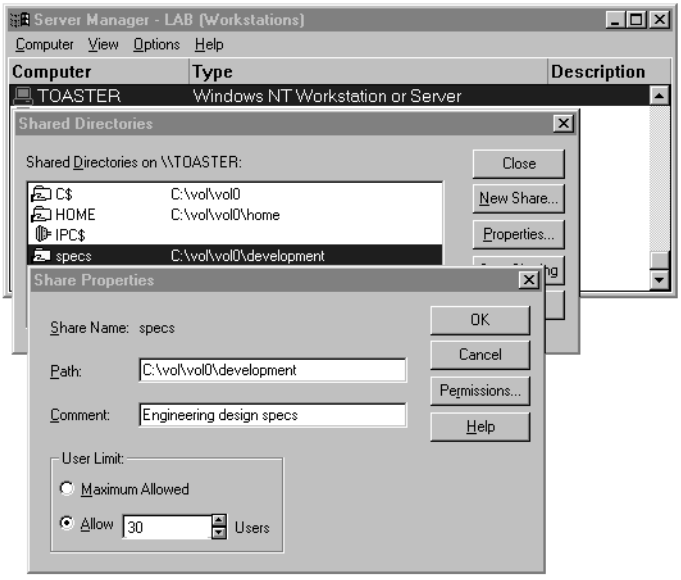
To create a share from the Data ONTAP command line, complete the following step.

Step	Action
1	Enter the following command: **cifs shares -add *shareName path* [-comment *description*] [-maxusers *userlimit*] [-forcegroup *groupname*] [-widelink] [-nosymlink_strict_security] [-novscan] [-novscanread] [-umask *mask*]** If an argument contains a space, enclose it in double quotation marks. Path separators can be backward or forward slashes, although Data ONTAP displays them as forward slashes. **Example:** `cifs shares -add webpages /vol/vol1/companyinfo -comment "Product Information" -forcegroup web -maxusers 100` **Result:** This command creates a share named webpages. The directory being shared is /vol/vol1/companyinfo. All files created by CIFS users in this share are owned by the web group, and a maximum of 100 users can access this share simultaneously.

Displaying and changing share properties

Displaying and changing share properties from Server Manager

To display and change share properties from Server Manager, complete the following steps.

Step	Action
1	Select your filer from Server Manager.
2	Display and change share properties as you would for a Windows NT server. **Example:** The following figure illustrates how to display the properties of a share named specs. In this example, the comment and the number of allowed users are changed. 

Displaying and changing share properties from the Data ONTAP command line

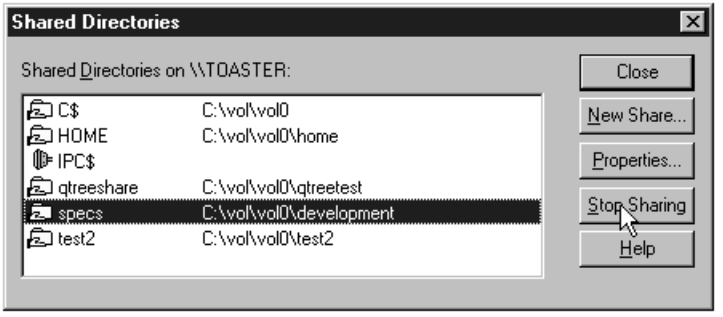
To display and change share properties from the Data ONTAP command line, complete the following steps.

Step	Action
1	Enter the following command: **cifs shares [*sharename*]** *sharename* is the name of a share you want to display. If you omit *sharename*, information about all shares is displayed. **Result:** For each share, this command displays the share name, the path name of the directory that is shared, the share description, and the share-level ACL.
2	Enter the following command: **cifs shares -change *sharename* { -comment *desc* | -nocomment } { -maxusers *userlimit* | -nomaxusers } { -forcegroup *groupname* | -noforcegroup } { -widelink | -nowidelink } { -symlink_strict_security | -nosymlink_strict_security } { -vscan | -novscan } { -vscanread | -novscanread } { -umask *mask* | -noumask }** Specifying `-nocomment`, `-nomaxusers`, `-noforcegroup`, and `-noumask` removes the description, the forcegroup, the maximum number of users, and the umask value, respectively, that you specified when you created the share.

Deleting a share

Deleting a share from Server Manager

To delete a share from Server Manager, complete the following steps.

Step	Action
1	Select your filer from Server Manager.
2	Delete a share as you would on a Windows NT server. **Example:** The following figure illustrates how to delete a share named specs. 

Deleting a share from the Data ONTAP command line

To delete a share from the Data ONTAP command line, complete the following step.

Step	Action
1	Enter the following command: **`cifs shares -delete [-f]`** ***`sharename`*** `-f` forces all files closed on a share without prompting. This is useful for scripts.

CIFS limitations

User Manager limitations

The following limitations apply to User Manager when you use User Manager for the filer:

- Although the filer supports local users, you cannot use the New Users command on the User menu to create or delete local user accounts.
- The Policies menu is disabled, but some policies can be controlled through options or group membership.

Server Manager limitations

The following Server Manager features are not supported because they are not applicable to Data ONTAP:

- Stopping and starting services
- Specifying the recipients of alerts

Limits for local user accounts

You can specify up to 96 local user accounts on the filer.

Limits for the F700 series filers

The following table shows access limits for the F700 series filers.

CIFS limits by filer memory	256 MB F720	512 MB F740	1024 MB F760
Maximum number of users	3,200	7,200	15,200
Maximum number of shares	6,400	14,400	30,400
Maximum number of open files	64,000	144,000	304,000
Maximum number of locked files	72,192	160,384	336,768
Maximum number of locks	144,384	320,768	673,536

Limits for the F800 series filers

The following table shows access limits for the F800 series filers.

CIFS limits by filer memory	512 MB F810	1GB F820	3 GB F840/F880
Maximum number of users	8,000	16,000	48,000
Maximum number of shares	16,000	32,000	96,000
Maximum number of open files	160,000	320,000	960,000
Maximum number of locked files	176,000	352,768	1,058,304
Maximum number of locks	352,768	705,536	2,116,608

Limits for the FAS900 series storage appliances

The following table shows access limits for the FAS900 series storage appliances.

CIFS limits by filer memory	3 GB FAS940	6 GB FAS960
Maximum number of users	48,000	96,000
Maximum number of shares	96,000	192,000
Maximum number of open files	960,000	1,920,000
Maximum number of locked files	1,058,304	2,116,608
Maximum number of locks	2,116,608	4,233,216

Limits for F85 and F87 filers

The following table shows access limits for F85 and F87 filers.

CIFS limits by filer memory	256 MB F85/F87
Maximum number of users	4,000
Maximum number of shares	8,000
Maximum number of open files	80,000
Maximum number of locked files	88,192
Maximum number of locks	176,384

Limits for the R100 storage appliance

The following table shows access limits for the R100 storage appliance.

CIFS limits by filer memory	3 GB
Maximum number of users	48,000
Maximum number of shares	96,000
Maximum number of open files	960,000
Maximum number of locked files	1,058,304
Maximum number of locks	2,116,608

Understanding local user accounts

Reasons for creating local users on the filer

There are several reasons for creating local user accounts on the filer:

- You must create local user accounts if, during setup, you configured the filer to be a member of a Windows workgroup. In this case, the filer must use the information in local user accounts to authenticate users.
- If your filer is a member of a domain:
  - Local user accounts enable the filer to authenticate users who try to connect to the filer from an untrusted domain.
  - Local users can access the filer when the domain controller is down or when network problems prevent your filer from contacting the domain controller. For example, you can define a BUILTIN\Administrator account so that you can use the account to access the filer even when the filer fails to contact the domain controller.

When you should not create local users

If, during setup, you configured the filer to use UNIX mode for authenticating users, do not create local user accounts. In UNIX mode, the filer always authenticates users using the UNIX password database.

Command for displaying the authentication method

The `cifs sessions` command displays the current user authentication method that the filer is configured to use.

Where local users can be used

Local users can be used in every case where you use user and group lists. For example, you can specify local users in file-level ACLs and share-level ACLs. You can also add local users to local groups.

Limitations of local user accounts

There are several limitations with local user accounts:

- You cannot use User Manager to manage local user accounts on the filer.
- You can use User Manager in Windows NT 4.0 only to view local user accounts. If you use User Manager in Windows 2000, however, you cannot

use the Users menu to view local users. You must use the Groups menu to display local users.

- You can create a maximum of 96 local user accounts.

Command for managing local users

The `useradmin` command creates, displays, and deletes local user accounts. You use the `useradmin` command as you do for creating, displaying, and deleting administrative users on the filer. For information about how to use the `useradmin` command, see the section about managing local user accounts in the introduction to filer administration in the *Storage Management Guide*.

Note

Data ONTAP keeps a single list of user accounts created by the `useradmin` command. The same types of information exist for local user accounts and administrative user accounts. That is, CIFS users who have local user accounts on the filer also have access to the filer console as administrative users.

Creating local groups on the filer

About local groups

You can define a local group on the filer that consists of users or global groups from any trusted domains. Members of a local group can be given access to files and resources.

Membership in certain well-known local groups confers special privileges on the filer. For example, members of BUILTIN\Power Users can manipulate shares, but have no other administrative capabilities.

How to create a local group

You can create a local group on the filer only from User Manager or Microsoft Management Console.

How clients display local groups

CIFS clients display the name of a local group in one of the following formats:

- *FILERNAME\localgroup*
- BUILTIN*localgroup*

How SnapMirror works with local groups

If you use the SnapMirror feature to copy a volume to another filer and the volume has an ACL for a local group, the ACL does not apply on the mirror. This is because the group is local to the source filer. Because the mirror is a read-only volume and you cannot change ACLs or permissions on it, do not use local groups in ACLs for files to be replicated by SnapMirror.

If you want to use local groups in ACLs for files to be replicated by SnapMirror, you can do this using the MultiStore™ product. For more information about the MultiStore product, see the *MultiStore Management Guide*.

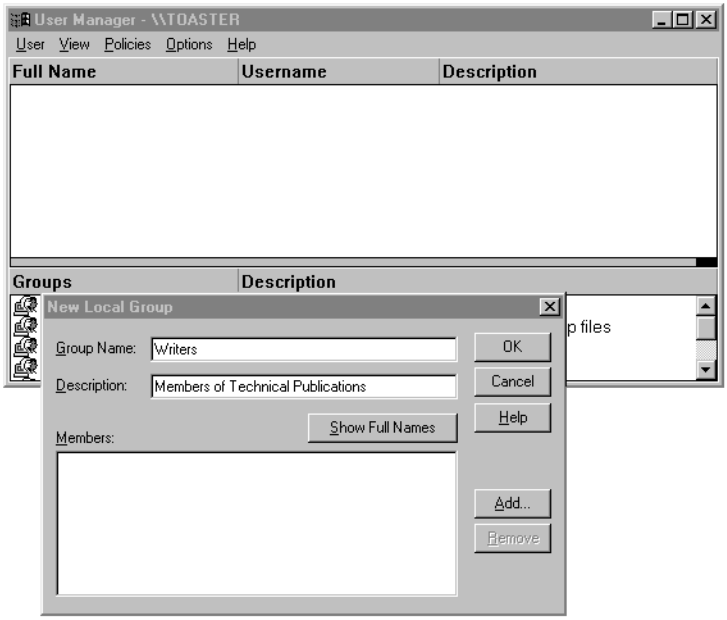
Creating a local group or adding users to a local group

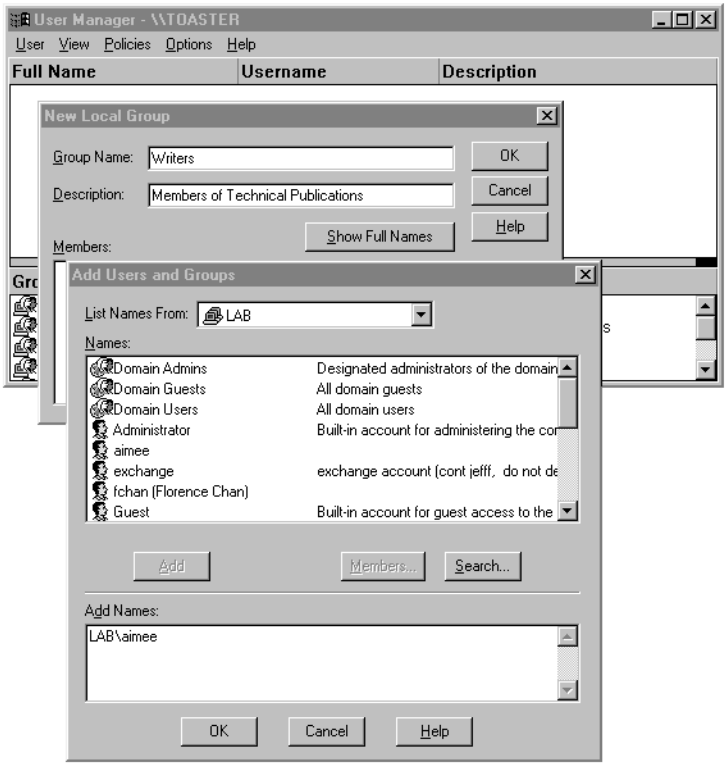
To create a local group or add users to a local group, complete the following steps.

Step	Action
1	From the User menu in User Manager, choose Select Domain.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>2</td><td colspan="2">In the Domain field, enter the following:
\\filer</td></tr>
<tr><td>3</td><td colspan="2">Click OK.</td></tr>
<tr><td rowspan="3">4</td><td>If you want to...</td><td>Then...</td></tr>
<tr><td>Create a local group</td><td>Go to Step 5.</td></tr>
<tr><td>Add members to an existing local group</td><td>Go to Step 7.</td></tr>
<tr><td>5</td><td colspan="2">From the User menu, choose New Local Group.</td></tr>
<tr><td>6</td><td colspan="2">Create a local group on the filer as you would on a Windows NT server.

Example: The following figure illustrates how to create a group named Writers.

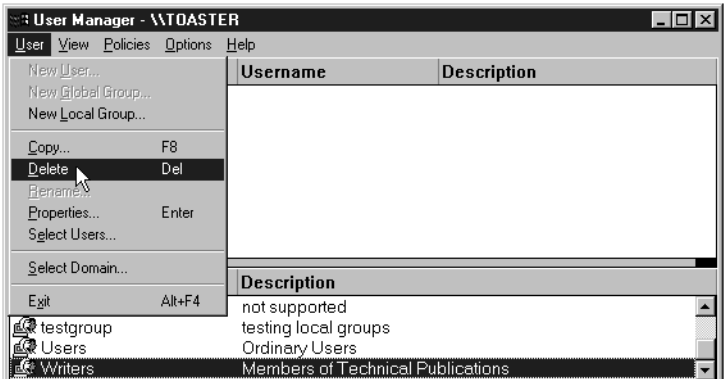
 
</td></tr>
</table>

Step	Action
7	Add users to the local group as you would on a Windows NT server. **Example:** The following figure illustrates how to add a user named aimee in the lab domain to the Writers group. 

Deleting a local group or removing members from a local group

To delete a local group or remove members from a local group, complete the following steps.

Step	Action
1	From the User menu in User Manager, choose Select Domain.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>2</td><td colspan="2">In the Domain field, enter the following:
\\filer</td></tr>
<tr><td>3</td><td colspan="2">Click OK.</td></tr>
<tr><td rowspan="3">4</td><td>If you want to...</td><td>Then...</td></tr>
<tr><td>Delete a local group</td><td>Go to Step 5.</td></tr>
<tr><td>Remove members from a local group</td><td>Go to Step 7.</td></tr>
<tr><td>5</td><td colspan="2">Select the local group you want to delete.</td></tr>
<tr><td>6</td><td colspan="2">From the User menu, choose Delete.
Example: The following figure illustrates how to delete a group named Writers.
 
</td></tr>
</table>

Step	Action
7	Remove members from a local group as you would on a Windows NT server. **Example:** The following figure illustrates how to remove the user named aimee in the lab domain from the Writers group. 

Understanding home directories on the filer

About home directories

You can create user home directories on the filer and configure Data ONTAP to automatically offer each home directory as a share to the user. The share is called the home share. From the CIFS client, the home share works as any other share to which the user can connect.

Each user can connect only to his or her home share, not home shares for other users.

How Data ONTAP matches users to directories

Data ONTAP offers the share to the user with a matching name. The user name for matching can be a Windows user name, a domain name followed by a Windows user name, or a UNIX user name. For more information about mapping users to directories, see "How Data ONTAP matches a directory with a user" on page 117.

Home directory names are not case-sensitive.

Where Data ONTAP searches for the home directories

When Data ONTAP tries to locate the directories named after the users, it searches only the paths that you specify. These paths are called home directory paths. They can exist in different volumes.

Example: If you specify /vol/vol1/enghome and /vol/vol2/mktghome as the home directory paths, Data ONTAP searches these paths to locate user home directories. If you create a directory for jdoe in the /vol/vol1/enghome path and a directory for jsmith in the /vol/vol2/mktghome path, both users are offered a home share. The home share for jdoe corresponds to the /vol/vol1/enghome/jdoe directory, and the home share for jsmith corresponds to the /vol/vol2/mktghome/jsmith directory.

Differences between a home share and other shares

The following differences exist between a home share and other shares:

- You cannot change the share-level ACL and the comment for a home share either from the Data ONTAP command line or from Server Manager.
- The `cifs shares` command does not display the home shares.

- The format of specifying the home share using the Universal Naming Convention (UNC) might be different from that for specifying other shares, as described in "Accessing home directories" on page 127.

How Data ONTAP matches a directory with a user

Effects of home directory naming styles on user matching

You can specify the naming style of home directories, which determines how Data ONTAP matches a directory with a user.

The following list describes the naming styles:

- Windows name: Data ONTAP searches for the directory whose name matches the user's Windows name.
- Windows domain name and Windows name: If users from different domains have the same user name, it is necessary to differentiate them using the domain name. If the naming style is Windows domain name and Windows name, Data ONTAP searches for a directory in the home directory path that matches the domain name. Then, in the domain directory, it searches for the home directory that matches the user name.

  Example: You can create a directory for engineering\jdoe and a directory for marketing\jdoe. To do so, you would create directories in the home directory paths. The directories created have the same names as the domain names (engineering and marketing). Then you would create user home directories in these domain directories.
- Mapped UNIX name: If the naming style is UNIX, Data ONTAP searches for the directory that matches the user's mapped UNIX name.

  Example: If John Doe's Windows name jdoe maps to the UNIX name johndoe, Data ONTAP searches the directory named johndoe, not jdoe, in the home directory paths, and offers it as the home share to John Doe.

Effects of not specifying a home directory naming style

If you do not specify a home directory naming style, Data ONTAP uses the pre-6.0 style, which is for backward compatibility. It is useful if you want Data ONTAP to match a directory with a user the same way it did before 6.0.

Before 6.0, Data ONTAP used the user's Windows name for directory matching. Therefore, the pre-6.0 style and the Windows naming style are similar. There are, however, differences between these styles, as described in "About symbolic links used as home directory names" on page 117.

About symbolic links used as home directory names

Before Data ONTAP 6.0, if you wanted home directories to reside in various volumes, you had to specify symbolic links as home directories in the home directory path. Because Data ONTAP 6.0 supports home directories in various

volumes, it is less likely that you need to use symbolic links as home directory names. However, Data ONTAP 6.0 continues to support symbolic links as home directory names for backward compatibility.

Effects of naming styles on symbolic links

The way symbolic links work depends on the home directory naming style.

Pre-6.0 naming style: If you do not specify a naming style, Data ONTAP uses symbolic links the same way it did before 6.0. That is, it can follow any symbolic link that points to a directory outside the home directory path to locate a home directory.

Example: Suppose the home directory path is /vol/vol0/eng_homes and you use the pre-6.0 home directory naming style. To locate the home directory for jdoe, Data ONTAP searches for /vol/vol0/eng_homes/jdoe, which can be a symbolic link pointing to a directory outside the home directory path, such as /vol/vol1/homes/jdoe.

Other naming styles: If you specify a home directory naming style, by default a symbolic link works only if the symbolic link points to a directory in the home directory path.

Example: Suppose the home directory path is /vol/vol0/eng_homes and you use the Windows naming style. To locate the home directory for jdoe, Data ONTAP searches for /vol/vol0/eng_homes/jdoe, which can be a symbolic link. If so, the user can access the home directory only if the target of the symbolic link resides in the home directory path. For example, the symbolic link works if it points to the /vol/vol0/eng_homes/john directory; it does not work if it points to the /vol/vol1/homes/john directory.

Note

You can change the default filer settings to allow CIFS clients to follow symbolic links to destinations outside the home directory path. For information on the options for symbolic links, see “Controlling CIFS access to symbolic links” on page 299.

Setting up user home directories

Tasks involved in setting up user home directories

The procedure for creating user home directories involves the tasks described in the following sections:

- "Specifying home directory paths" on page 120
- "Specifying the naming style of home directories" on page 124
- "Creating directories in a home directory path" on page 125

Specifying home directory paths

About home directory paths

You can specify multiple home directory paths. Data ONTAP searches the home directory paths in the order you specify them for the directory that matches the user name. It stops searching after the matching directory is found.

About home directory path extensions

You can add an extension to the home directory path if you do not want users to access the top level of their home directories. The extension specifies the subdirectory that is automatically opened when users access their home shares. An example of adding an extension is shown in "Specifying home directory paths from the command line" on page 120.

Ways to specify home directory paths

You can specify home directory paths in either of the following ways:

- Using the command `options cifs.home_dir`
- Using the /etc/cifs_homedir.cfg file

The `options cifs.home_dir` command allows you to specify home directory paths as a comma separated list. The list has a limit of 255 characters, including the commas. Alternatively, you can specify up to 1,000 path names in the /etc/cifs_homedir.cfg file.

Where Data ONTAP searches for home directory paths

Data ONTAP searches first for home directory paths specified using the `options cifs.home_dir` command. If Data ONTAP finds paths specified from the command line, Data ONTAP ignores any paths in the cifs_homedir.cfg file. If there are no existing paths that were specified using `options cifs.home_dir`, Data ONTAP searches for home directory paths in the cifs_homedir.cfg file.

Specifying home directory paths from the command line

To create home directory paths from the command line, complete the following steps.

Step	Action
1	Create directories to use as home directory paths. **Example:** In the /vol/vol0 volume, create a directory named enghome, and in the /vol/vol1 volume, create a directory named mktghome.
2	Enter the following command to designate the directories created in Step 1 as the paths where Data ONTAP searches for user home directories: **`options cifs.home_dir`** ***`pathname`*****`,...`** **Example 1:** `options cifs.home_dir /vol/vol0/enghome,/vol/vol1/mktghome` **Result:** When a CIFS user connects to the filer, Data ONTAP searches the /vol/vol0/enghome path and the /vol/vol1/mktghome path for the user's home directory. If found, it offers the directory as a share to the user. If the path name includes a home directory path extension, use the following format: *`home_dir_path`*`/%u%/`*`extension`* *home_dir_path* is the home directory path. *extension* is the name of the subdirectory that you want users to access. **Example 2:** `options cifs.home_dir /vol/vol0/enghome/%u%/data` **Result:** When a CIFS user connects to the filer, Data ONTAP searches the /vol/vol0/enghome path for the user's home directory. If found, it offers the data directory in the user's home directory as the home share. The share name does not reflect the extension; it is the same as if an extension were not used. **Note** You can enter home directory paths up to a limit of 255 characters, including the commas.

About the default cifs_homedir.cfg file

Data ONTAP creates a default cifs_homedir.cfg file in the /etc directory when CIFS starts, if the file does not already exist. Changes to this file are processed automatically whenever CIFS starts. You can also process changes to this file using the command `cifs homedir load`.

Specifying home directory paths in cifs_homedir.cfg

To specify home directory paths in the /etc/cifs_homedir.cfg file as the paths where Data ONTAP searches for user home directories, complete the following steps.

Step	Action
1	Create directories to use as home directory paths. **Example:** In the /vol/vol0 volume, create a directory named enghome.
2	Open the /etc/cifs_homedir.cfg file for editing.
3	Enter the home directory path names created in Step 1 in the /etc/cifs_homedir.cfg file, one entry per line, to designate them as the paths where Data ONTAP searches for user home directories. **Note** You can enter up to 1,000 path names.
4	Enter the following command to process the entries: **`cifs homedir load [-f]`** `-f` forces the use of the new paths.

Effect of changing home directory paths with open files

You can change the home directory paths at any time using either the `options cifs.home_dir` command or by changing the entries in the cifs_homedir.cfg file. However, if a user has open files in a home directory path that you remove from the list, Data ONTAP displays a warning message and requests a confirmation for the change. If you use the `-f` flag with the `cifs homedir load` command, you will not receive a warning message before the new path names take effect. Changing a directory path that contains an open file terminates the connection to the home directory.

Displaying the list of home directory paths

To display the current list of home directory paths, complete the following step.

Step	Action
1	Enter the following command: `cifs homedir`

Specifying the naming style of home directories

Specifying the naming style of home directories

To specify the name style used for matching home directories to users, complete the following step.

Step	Action
1	Enter the following command: **`options cifs.home_dir_namestyle {ntname|domain|mapped|""}`** Use `ntname` if the home directories have the same names as the Windows user names. Use `domain` if you want to use the domain name in addition to the Windows user name to search for the home directory. Use `mapped` if the home directories have the UNIX user names as specified in the usermap.cfg file. Use `""` if you do not want to specify a name style and want Data ONTAP to match home directories to users the same way it did before Data ONTAP 6.0. By default, the `cifs.home_dir_namestyle` option is `""`.

Creating directories in a home directory path

Creating home directories if home directory naming style is domain

To create home directories if the `cifs.home_dir_namestyle` option specifies `domain`, complete the following steps.

Step	Action
1	In one of the specified home directory paths, create directories that will contain home directories used by all users of the same domain. For each directory, use the same name as the domain. **Example:** If there are two domains, engineering and marketing, create the /vol/vol0/enghome/engineering directory and the /vol/vol1/mktghome/marketing directory.
2	In each domain directory created in Step 1, create home directories to be used by the users in the same domain. **Example:** If two users are both named jsmith and they are in the engineering domain and the marketing domain, create the /vol/vol0/enghome/engineering/jsmith home directory and the /vol/vol1/mktghome/marketing/jsmith home directory.
3	Make each user the owner of his or her home directory. **Example:** Make engineering\jsmith the owner of the /vol/vol0/enghome/engineering/jsmith home directory and marketing\jsmith the owner of the /vol/vol1/mktghome/marketing/jsmith home directory. **Result:** Engineering\jsmith can attach to the share named jsmith, which corresponds to the /vol/vol0/enghome/engineering/jsmith home directory; marketing\jsmith can attach to the share named jsmith, which corresponds to the /vol/vol1/mktghome/marketing/jsmith home directory.

Creating home directories if home directory naming style is not domain

To create home directories if the `cifs.home_dir_namestyle` option specifies `ntname`, `mapped`, or `""`, complete the following steps.

Step	Action
1	In one of the specified home directory paths, create home directories. **Example:** If there are two users, jsmith and jdoe, create the /vol/vol0/enghome/jsmith and /vol/vol1/mktghome/jdoe home directories. **Result:** Users can attach to the share that has the same name as their user name and start using the share as their home directory.
2	Make each user the owner of the home directory. **Example:** Make jsmith the owner of the /vol/vol0/enghome/jsmith home directory and jdoe the owner of the /vol/vol1/mktghome/jdoe home directory. **Result:** Engineering\jsmith can attach to the share named jsmith, which corresponds to the /vol/vol0/enghome/engineering/jsmith home directory; marketing\jsmith can attach to the share named jsmith, which corresponds to the /vol/vol1/mktghome/marketing/jsmith home directory.

Creating subdirectories in home directories when a home directory path extension is used

To create subdirectories that users can access in their home directories if you use a home directory path extension, complete the following step.

Step	Action
1	For each home directory that resides in a home directory path with an extension, create a subdirectory that you want users to access. **Example:** If the `cifs.home_dir` option specifies the /vol/vol0/enghome/%u%/data path, create a subdirectory named data in each home directory. **Result:** Users can attach to the share that has the same name as their user name. When they read or write to the share, they effectively access the data subdirectory.

Accessing home directories

Attaching to one's own home share through Network Neighborhood

Attaching to the home share through Network Neighborhood is the same as attaching to any other share.

Syntax for specifying a home share using a UNC name

You can also access your home share by using a UNC name. The convention for specifying a home share when using the UNC depends on the home share name style specified by the `cifs.home_dir_namestyle` option. The following table describes the different UNC names.

Value of cifs.home_dir_namestyle	UNC name
`ntname` or `""`	*filer**Windows_NT_name* **Example:** `\\toaster\jdoe`
`domain`	*filer*\~*domain*~*Windows_NT_name* **Example:** `\\toaster\~engineering~jdoe`
`mapped`	*filer*\~*mapped_name* **Example:** `\\toaster\~jdoe`

If `cifs.home_dir_namestyle` is `domain` but the UNC name in the access request does not specify a domain name, Data ONTAP assumes the domain to be the domain under which the request is sent. If you omit the domain name in the access request, you can also leave out the tilde (~) before the user name.

Example: A user named jdoe is logged in as engineering\jdoe from a PC in the engineering domain. Suppose he tries to access his home directory, using his user name in the marketing domain. He can enter either of the following commands to request access to his home directory:

- `net use * \\toaster\~jdoe /user:marketing\jdoe`
- `net use * \\toaster\jdoe /user:marketing\jdoe`

Data ONTAP assumes the domain to be marketing, not engineering, because marketing is the domain of marketing\jdoe, who sends the request.

Enabling users to access other users' home directories

A user can attach to his or her own home share, not the home shares of other users. If you want users to be able to access all other users' home shares, complete the following steps.

Step	Action
1	Create a share as you usually would that corresponds to the path name that is either ◆ A home directory path if `cifs.home_dir_name_style` is not `domain` ◆ A domain directory in the home directory path if `cifs.home_dir_name_style` is `domain` **Example:** If /vol/vol0/enghome is a home directory path, use the following command: `cifs shares -add eng_dirs /vol/vol0/enghome -comment "readable engineering home directories"`
2	Assign each user the appropriate access permissions to other users' home directories. **Example:** Assign read-only permission to the engineering group for the eng_dirs share as follows: `cifs access eng_dirs engineering full` **Result:** Members of the engineering group have read-only access to all home directories in the eng_dirs share.

Accessing your CIFS home share using a share alias

For any CIFS home share name style, you can connect to your own CIFS home share using either of the following share aliases:

- cifs.homedir
- ~

Examples:

```
net use * \\toaster\cifs.homedir
```

```
net use * \\toaster\~
```

This can be useful when you are writing scripts.

Specifying support for wide symbolic links in CIFS home directories

Specifying support for wide symbolic links in home directories

You can specify support for wide symbolic links in all users' home directory shares using cifs.homedir as the share name for home directories. When you enable wide symbolic links in home directory shares, CIFS clients can follow wide symbolic links in all home directory shares to destinations in the same share or outside the share.

To change support for wide symbolic links in CIFS home directory shares, complete the following step.

Step	Action
1	Enter the following command: **`cifs shares -change cifs.homedir [ -widelink | -nowidelink]`** **Result:** All user's home directory shares are enabled or disabled for wide symbolic link support.

For more information

For more information about the `cifs.homedir` option, see "Specifying home directory paths" on page 120. For details about wide symbolic links, see "About Widelink entries" on page 304 and "Sharing directories on the filer" on page 95.

How to stop offering home shares

How to stop offering home shares

You cannot use the `cifs shares -d` command to delete home shares. To stop offering home shares you can either delete the /etc/home_dir file or complete the following step.

Step	Action
1	Enter the following command: **`options cifs.home_dir ""`** **Result:** Data ONTAP stops offering home shares to CIFS users.

How share-level access control lists work

About share-level ACLs

A share-level ACL (access control list) consists of a list of access control entries (ACEs). Each ACE contains a user or group name and a set of permissions that determines user or group access to the share.

A share-level ACL only restricts access to files in the share; it never grants more access than the file-level ACLs.

When a CIFS user tries to access a share, Data ONTAP always checks the share-level ACL to determine whether access should be granted, regardless of the security style of the qtree containing the share.

Permission styles and types in a share-level ACL

When changing a share-level ACL, you can specify Windows permissions or UNIX-style permissions.

Note

If you use UNIX-style permissions, you cannot use Server Manager to manage the share-level ACL.

The following table describes the permissions you can specify depending on the permission style.

Permission style	Permissions
Windows permissions	◆ No Access ◆ Read ◆ Change ◆ Full Control
UNIX permissions	Combinations of r, w, x, and -.

Specifying how GIDs work with share-level ACLs

Reason for deciding whether GID controls group access

This section applies only if you are using UNIX share-level access. If a share contains files with UNIX-style security and if you want to use the share-level ACL to control access by UNIX groups, you must decide whether Data ONTAP grants user access to files based on GID. This is necessary because of the differences between UNIX and Windows security policies involving group permissions.

Example

A share named specs exists in a UNIX-style qtree. Because you believe two UNIX groups, engineering and marketing, should have full access to the share, you give rwx permissions to these groups at the share level.

Suppose in this share, a file owned by the engineering group is named draft and it has the following permissions:

```
draft rwxr-x---
```

When a member of engineering tries to access the draft file, the share-level ACL works properly. That is, this user has unrestricted access to the specs share, and access to the draft file is determined by the access rights assigned to the engineering group, which is r-x.

However, when a member of marketing tries to access the draft file, access is denied because, according to the UNIX-style file permissions, nonmembers of engineering should have no access to the file. Making the draft file readable by the marketing group requires that you change the file-level permissions to the following settings:

```
draft rwxr-xr-x
```

The disadvantage of these permissions is that while marketing can read the file, so can all UNIX users, which creates a security problem.

To solve this problem, you can configure Data ONTAP to disregard the GID when granting access.

Effect of disregarding the user's GID

If you configure Data ONTAP not to consider the user's GID when granting access, all users who are not the file owner are considered members of the UNIX group that owns the file. In the preceding example, permissions that apply to the

engineering group also apply to members of marketing who try to access the file. That is, both engineering members and marketing members have the r-x permissions to the draft file.

When to specify that the user's GID should be considered

By default, Data ONTAP considers the user's GID before granting access. This default configuration is useful if either of the following statements is true:

- The share does not contain files with UNIX-style security.
- You do not use a share-level ACL to control any UNIX group's access.

Specifying whether GID affects file access

To specify whether the user's GID affects file access, complete the following steps.

Step	Action	
1	**If you want Data ONTAP to...**	**Then...**
	Disregard the GID when granting user access	Complete Step 2.
	Consider the GID when granting user access	Complete Step 3.
2	Enter the following command: `options cifs.perm_check_use_gid off`	
3	Enter the following command: `options cifs.perm_check_use_gid on`	

Changing and displaying a share-level ACL

Reason for changing a share-level ACL

After you create a share, by default, the share-level ACL gives Full Control to the special group named Everyone. That is, all users in the domain and all trusted domains have full access to the share; their access to the share contents is restricted only by file- and directory-level ACLs. You would change the share-level ACL if you want to use a more restrictive set of permissions for the share.

Methods of changing a share-level ACL

To change a share-level ACL, use one of the following methods:

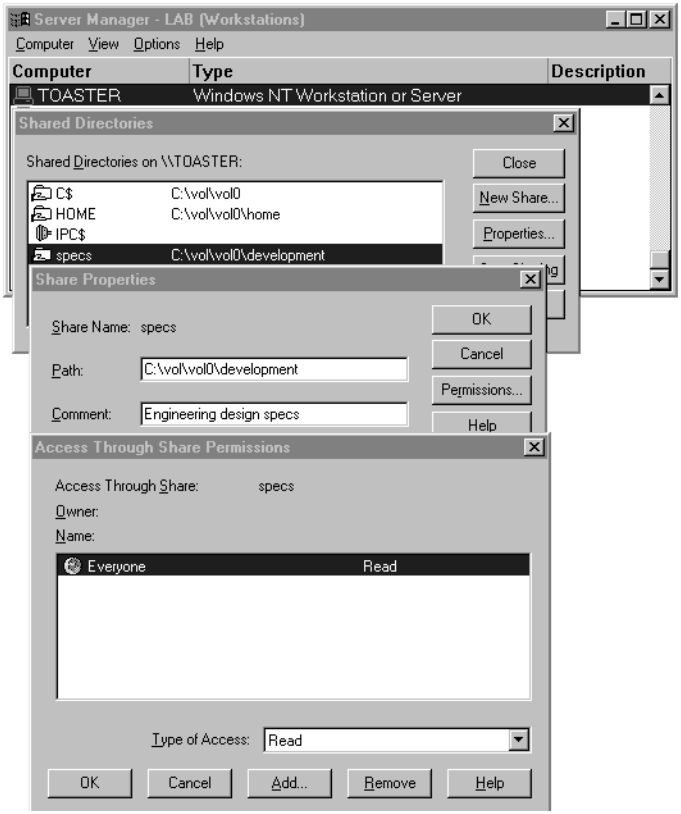
- From a Windows client, use Server Manager. Remember these guidelines if you use this method:
  - You can specify only Windows permissions.
  - The user and group names specified must be Windows names.
  - The share-level ACL must not have UNIX-style permissions; otherwise, Server Manager displays an error message.
- From the Data ONTAP command line, you can use the `cifs access` command. Remember these guidelines if you use this method:
  - You can specify either Windows permissions or UNIX-style permissions.
  - The user and group names can be Windows or UNIX names.

    If the filer is authenticated by the /etc/passwd file, the user or group name in the ACL is assumed to be a UNIX name. If the filer is authenticated by a domain controller, the name is assumed to be a Windows name, unless the name is not found on the domain controller. In that case, the filer tries to look up the name in the UNIX name database.

Displaying and changing an ACL with Server Manager

To display and change an ACL with Server Manager, complete the following steps.

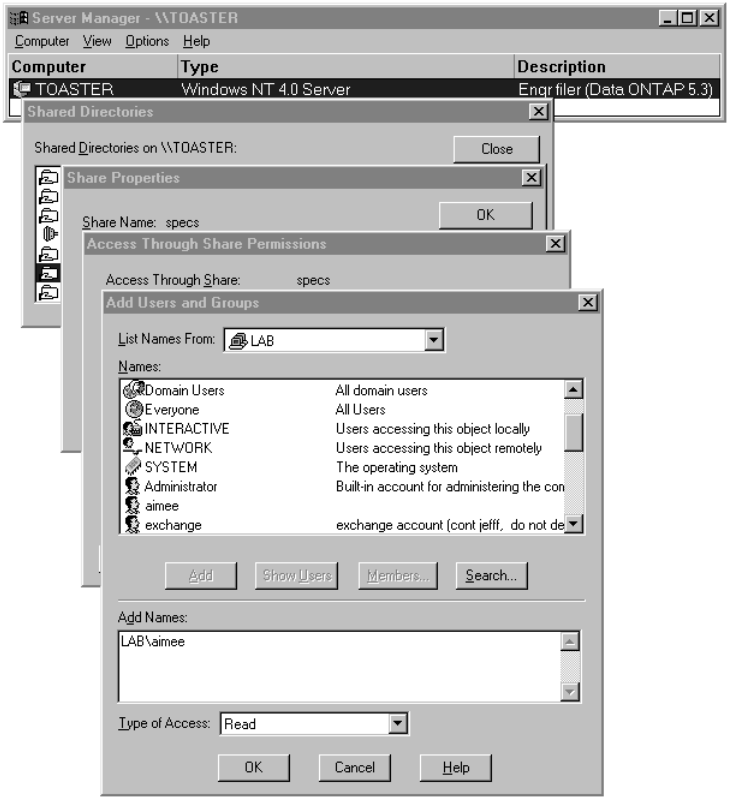
Step	Action
1	Select your filer from Server Manager.

Step	Action
2	Display and change the permissions for a selected share as you would on a Windows NT server. **Example of changing an existing ACL:** The following figure illustrates how to display and then change the permissions for the specs share so that the Everyone group can read it. 

Adding an entry to an ACL with Server Manager

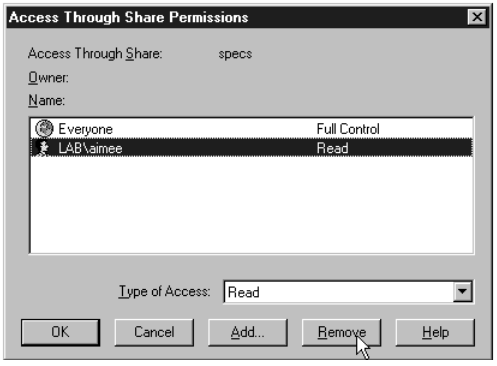
To add an entry to an ACL with Server Manager, complete the following steps.

Step	Action
1	Select your filer from Server Manager.

Step	Action
2	Add a user or group to the ACL as you would on a Windows NT server. **Example of adding an entry to an ACL:** The following figure illustrates how to add a user named LAB\aimee to the ACL for the specs share. 

Removing an entry from an ACL with Server Manager

To remove an entry from an ACL with Server Manager, complete the following steps.

Step	Action
1	Select your filer from Server Manager.
2	Remove a user or group from the ACL as you would on a Windows NT server. **Example of removing an entry from an ACL:** The following figure illustrates how to remove a user named LAB\aimee from the ACL for the specs share. 

Changing an ACL from the Data ONTAP command line

To change an ACL from the Data ONTAP command line, complete the following step.

Step	Action
1	Enter the following command: **`cifs access share [ -g ] { user | group } rights`** The `-g` argument specifies that the permissions are defined for a UNIX group. Do not include the `-g` argument if the permissions are for a Windows group. The command syntax does not provide a mechanism to differentiate between a Windows user name and a Windows group name. *user* is a UNIX user name or a Windows user name. *group* is a UNIX group name or a Windows group name. If the group is a local group, specify the filer name as the domain name (for example, toaster\writers). *rights* is a set of UNIX-style permissions or Windows permissions. **Examples:** The following examples assign access rights to the share named library: ◆ `cifs access library -g engineering rwx` ◆ `cifs access library marketing\joed Change`

Removing an entry from an ACL with the Data ONTAP command line

To remove an entry from an ACL with the Data ONTAP command line, complete the following step.

Step	Action

1	Enter the following command:
	`cifs access -delete` ***`share`*** **`{`** ***`user`*** **`|`** ***`group`*** **`}`**
	Examples: The following command removes the engineering group from the ACL for the library share:
	`cifs access -delete library engineering`
	The following command removes the user marketing\joed from the ACL for the library share.
	`cifs access -delete library marketing\joed`

Displaying share-level ACLs from the Data ONTAP command line

To display share-level ACLs, complete the following step.

Step	Action
1	Enter the following command: **`cifs shares [`** ***`share_name`*** **`]`** **Result:** If the filer uses password authentication, the permission style displayed in share-level ACLs is UNIX style. If the filer uses domain controller authentication, the permission style is identical to the style you used when specifying the ACLs, with the following exceptions: ◆ If you specified rwx as the UNIX-style permissions, the permissions are displayed as Full Control in the ACL. ◆ If you specified --- as the UNIX-style permissions, the permissions are displayed as No Access in the ACL. ◆ If you specified r-x as the UNIX-style permissions, the permissions are displayed as Read in the ACL.

Changing and displaying file-level ACLs

About file-level ACLs

Permission settings for files and directories are stored in file-level ACLs. These ACLs follow the Windows 2000 NTFS security model. For files that have NTFS-style security, CIFS users can set and display file-level ACLs from their PC. All files in an NTFS-style qtree and some files in a mixed qtree can have NTFS-style security. Files in a UNIX qtree do not have ACLs; they use UNIX permissions.

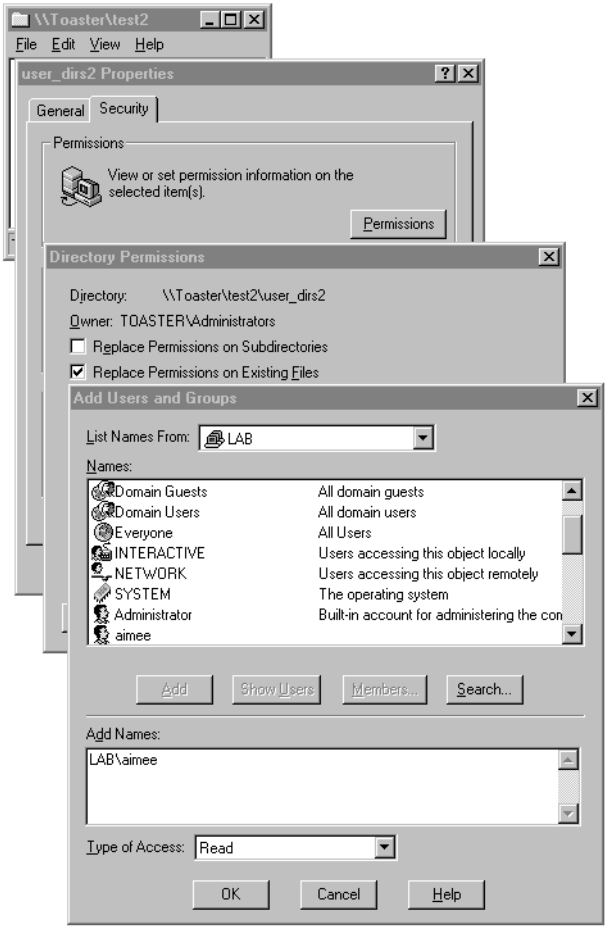
Permission types in a file-level ACL

Data ONTAP supports all permission types supported by Windows file-level ACLs.

Setting and displaying a file-level ACL

To set and display a file-level ACL, complete the following steps.

Step	Action
1	From the Windows desktop, right-click a file and select Properties from the pop-up menu.
2	Click the Security tab.

Step	Action
3	Click Permissions. **Example:** The following figure illustrates how to change the file-level ACL for a folder in the test2 share by adding LAB\aimee to the ACL. 

Note

On an NT4 client, if you right-click a file that is located in a share that supports wide symbolic links and select Properties, no Security tab is displayed. You can set security using a security tool such as cacls. Alternatively, you can either access files from a Windows 2000 client or access files using shares that don't support wide symbolic links. You can have two different shares on the same directory, one that supports wide symbolic links and one that does not, and use the share that does not support wide symbolic links when setting security.

Understanding the SID-to-name map cache

Purpose of the SID-to-name map cache

CIFS frequently is required to map security identifiers (SIDs) to user and group names and vice versa for user authentication, quota management, console command processing, and various RPC responses. The filer obtains the SID-to-name mapping information by querying the domain controller. To minimize multiple lookups of the same names, SID-to-name information received from the domain controller is saved in the SID-to-name map cache on the filer.

What the cache contains

The SID-to-name map cache contains entries that map SIDs to pre-Windows 2000 user and group names. SID-to-name mapping entries have a limited lifetime.

How the cache is controlled

The SID-to-name map cache is enabled on the filer by default. You can manually control the cache by changing the lifetime of the entries, clearing entries, or turning SID-to-name map caching off or on. The cache persists if CIFS is terminated or restarted, but does not persist across a reboot or a takeover and giveback.

How the filer uses the cache

When the filer requires SID-to-name mapping information, it first looks for a matching entry in the SID-to-name map cache. If a matching entry is not found or if an expired matching entry is found, the filer queries the appropriate domain controller for current mapping information. If the domain controller is not available, an expired mapping entry may be used by the filer.

Benefits of using the cache

The main benefits of using the SID-to-name map cache for name lookup are

- Increased performance for authorization
- Faster user response for console commands that perform mapping operations

Managing the cache

The following sections discuss how you can manage the SID-to-name map cache:

- "Enabling and disabling the SID-to-name map cache" on page 146

- "Changing the lifetime of SID-to-name mapping entries" on page 147
- "Clearing the SID-to-name map cache" on page 148

Enabling and disabling the SID-to-name map cache

Enabling and disabling the cache

To enable or disable caching of SID-to-name translation information that CIFS receives from domain controllers, complete the following step.

Step	Action
1	Enter the following command: `options cifs.sidcache.enable on | off`

Changing the lifetime of SID-to-name mapping entries

Changing the lifetime of mapping entries

The default lifetime of SID-to-name mapping entries is 1440, which is 24 hours. Lifetime is expressed in minutes.

To change the lifetime of mapping entries, complete the following step.

Step	Action
1	Enter the following command: **`options cifs.sidcache.lifetime`** ***`time`*** *time* represents the number of minutes that new mapping entries are used before they expire.

Clearing the SID-to-name map cache

Reason for manually clearing the cache

You may want to manually clear the SID-to-name map cache when a user changes their account or user name.

Automatic clearing of expired entries

Periodically, expired entries that are more than one week old are automatically cleared from the SID-to-name map cache.

Clearing all cache entries

To clear all SID-to-name map cache entries, complete the following step.

Step	Action
1	Enter the following command: **`cifs sidcache clear all`**

Clearing the cache entries for a domain

To clear the cache entries for a specific Windows domain, complete the following step.

Step	Action
1	Enter the following command: **`cifs sidcache clear [ domain ]`** *domain* is the Windows domain of the cache entries you want to clear. If you do not specify the domain, entries for the filer's home domain are cleared from the cache.

Clearing the cache entry for a user or group

To clear the cache entry for a specific user or group, complete the following step.

Step	Action
1	Enter the following command: **`cifs sidcache clear user username`** *username* is the specific Windows user or group entry you want to clear from the cache. The user name can be specified in the following ways: ◆ *domain\username* ◆ *username* When the user name is specified without a domain, the filer's home domain is used for the domain.

Clearing the cache entry for an SID

To clear the cache entry for a specific SID, complete the following step.

Step	Action
1	Enter the following command: **`cifs sidcache clear sid textualSid`** *textualSid* is the textual form of the SID you want to clear from the cache. Specify the SID using standard "S-1-5..." syntax. **Example:** S-1-5-21-4503-17821-16848-500

Understanding file screening

File screening in Data ONTAP

File screening in Data ONTAP is the coordination of a Data ONTAP file screening policy with file screening software. The Data ONTAP file screening policy is set on a filer and specifies the types of CIFS files you want to screen. The file screening software runs on a client that functions as a file screening server.

Prerequisite for file screening

Before you use file screening, you need to obtain client file screening software that works with Data ONTAP. You can check for client file screening software on the NetApp Partners page at http://www.netapp.com/partners/.

About file policies

A file policy determines how the filer handles requests from individual client systems for operations such as open, rename, and create. The filer maintains a set of properties for a file policy, including, for example, the policy name and whether the policy is active. You can set these properties for the file policy using console commands on the filer.

The only file policy supported on NetApp filers is file screening for CIFS access. The only policy name supported is “default.”

How file screening works

You use the file screening policy to specify that certain types of CIFS files are screened before they can be opened, created, or renamed. For example, you might use file screening to block users from creating audio or graphics files by specifying that audio and graphics files be screened.

Once your policy is set, when the filer receives a request to open, create, or rename a file, the filer checks its file screening policy. If the policy specifies screening for that file based on its extension, the file name is sent to the file screening server to be screened. The file screening server applies policies to the file name to determine whether the filer should allow the requested file operation. The file screening server then sends a response to the filer to either allow or block the requested file operation.

Enabling and disabling file screening

Enabling and disabling all file policies

All file policies are enabled by default. To enable or disable all file policies, complete the following step.

Step	Action
1	Enter the following command: **`options fpolicy.enable [on | off]`**

Enabling a specific file policy

To enable a specific file policy, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy enable`** ***`PolicyName`*** Use `default` as the policy name for the default NetApp file screening policy. Currently, `default` is the only supported policy name. **Example:** **`fpolicy enable default`** **Note** To activate file screening, make sure that `fpolicy` is enabled and `options fpolicy.enable` is turned on. In addition, a file screening server must be available.

Disabling a specific file policy

To disable a specific file policy, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy disable PolicyName`** Use `default` as the policy name for the default NetApp file screening policy. Currently, `default` is the only supported policy name. **Example:** **`fpolicy disable default`**

Managing file policies

Displaying file policy information

To display file policy information, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy show PolicyName`** Use `default` as the policy name for the default NetApp file screening policy. Currently, `default` is the only supported policy name.

Example of file screening policy information display

Following is an example of the information display for a file policy.

```
File policy default file screening is enabled.

File screen servers      Connect time   Reqs   Fails
-------------------------------------------------------
10.34.18.70 \\SCREEN1      00:00:05     1032     0

Operations monitored: File Open, File Create, File Rename.

List of extensions not to screen:
EXE, DOC

Number of files screened: 1032

Number of failures: 0
```

How the file screening policy specifies which files to screen

The file screening policy specifies which files to screen using a list of file extensions to exclude from screening. The default list of excluded file extensions is empty. This means that by default all files with any type of extension are screened by the file screening server before they can be opened, created, or renamed.

From the command line, you can display or change the list of excluded file extensions.

Displaying the list of excluded file extensions

To display the list of excluded file extensions, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy ext[ensions] excl[ude] show`** ***`PolicyName`*** Use `default` as the policy name for the default NetApp file screening policy. Currently, `default` is the only supported policy name.

Excluding file extensions from file screening

To add file extensions to the list of excluded file extensions, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy ext[ensions] excl[ude] add`** ***`PolicyName ext`*****`[,`*****`ext`*****`]...`** Use `default` as the policy name for the default NetApp file screening policy. Currently, `default` is the only supported policy name. **Example:** **`fpolicy ext excl add default doc,exe,txt`** **Result:** Doc, exe, and txt are added to the current list of extensions not to screen. **Note** The default list of excluded extensions is empty, so all files are screened by default.

Removing file extensions from the list of excluded extensions

To remove file extensions from the list of excluded file extensions, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy ext[ensions] excl[ude] remove PolicyName ext[,ext]...`** Use `default` as the policy name for the default NetApp file screening policy. Currently, `default` is the only supported policy name. **Example:** **`fpolicy ext excl remove default wav`** **Result:** Files with a .wav extension are screened.

Resetting default file screening of all files

To reset to the default file screening of all files, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy ext[ensions] excl[ude] reset PolicyName`** Use `default` as the policy name for the default NetApp file screening policy. Currently, `default` is the only supported policy name. **Result:** The list of excluded file extensions is reset to the default empty list, so no file extensions are exempted; therefore, all files are screened.

Replacing the list of excluded file extensions

To replace the entire existing list of excluded file extensions, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy ext[ensions] excl[ude] set PolicyName ext[,ext...]`** Use `default` as the policy name for the default NetApp file screening policy. Currently, `default` is the only supported policy name. **Result:** The new list of extensions you enter with this command replaces the existing list of excluded extensions so that only the new extensions are excluded from screening. **Note** If you don't want any files to be screened, use ??? for *ext* as a wildcard for "all extensions" or, alternatively, you can disable the file screening policy.

Requiring file screening for file access

To require files to be screened before they can be accessed, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy options default required on`** **Note** If you turn on required file screening and no file screening servers are available, file access will be denied. This command is set to Off by default, that is, the default file policy does not require file screening for file access.

Managing file screening servers

Displaying file screening server status

To display the status of file screening servers, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy servers show`** ***`PolicyName`*** Use `default` as the policy name for the default NetApp file screening policy. Currently, `default` is the only supported policy name.

Disabling the connection to a file screening server

To disable the connection to a file screening server, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy servers stop`** ***`PolicyName IPaddr`*** *IPaddr* is the IP address of the file screening server. Use `default` as the policy name for the default NetApp file screening policy. Currently, `default` is the only supported policy name.

Sending a message to all users on a filer

Purpose of sending a message

You send a message to all users on a filer to tell them of important events. The message appears in an alert box on the users' computers.

Data ONTAP automatically sends a message to connected users after you enter the `cifs terminate` command. However, if you want to send a message without stopping CIFS service, for example, to tell users to close all files, use Server Manager or the Data ONTAP command line to send a message.

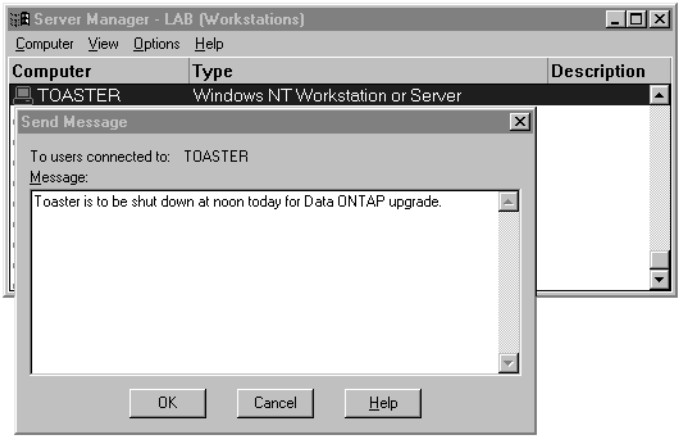
Prerequisite for displaying the message

Windows 95 and Windows for Workgroups clients must have the WinPopup program configured before they can display any messages you send from the filer.

Sending a message to all connected users from Server Manager

To use Server Manager to send a message to all users on a filer, complete the following steps.

Step	Action
1	Select your filer from Server Manager.

Step	Action
2	From the Computer menu, choose Send Message as you would when sending a message to users on a Windows NT server. **Example:** The following example illustrates how to send a message to all users on toaster. 

Sending messages to connected users from the command line

To use the Data ONTAP command line to send a message to one client or to all clients connected to a particular volume, complete the following steps.

Step	Action	
1	**If you want to...**	**Then...**
	Send a message to all CIFS users connected to the filer	Go to Step 2.
	Send a message to all CIFS users connected to a particular volume	Go to Step 3.
2	Enter the following command: **`cifs broadcast client_name "message"`**	

Step	Action
3	Enter the following command **cifs broadcast -v *volume* "*message*"**

Understanding CIFS auditing

About CIFS auditing

Data ONTAP audits logon, logoff, and file access events similarly to Windows. There are some differences, however, in how you enable auditing and how you configure the file that logs audit event information.

Events that Data ONTAP can audit

You can enable auditing for the following categories of events:

- Logon and logoff events
- File access events

Prerequisites for auditing file access events

Following are the prerequisites for auditing file access events:

- The file or directory to be audited must be in a mixed or NTFS volume or qtree. You cannot audit events for a file or directory in a UNIX volume or qtree.
- You must activate access auditing for individual files and directories according to your Windows documentation.

Where Data ONTAP logs audit event information

Audit event information is stored in an internal log file, /etc/log/cifsaudit.alf. You should periodically save the contents of this file to an external event log file either manually or by setting up automatic saving of this file. By default, the external event log is the /etc/log/adtlog.evt file. You can specify another file as the event log. The specified file does not need to be an existing file. Data ONTAP creates the file when it saves information to the file. The directory containing the file, however, must exist; otherwise, an error appears after you specify the file.

Size and format of the internal and external log files

You can specify the maximum size of the internal cifsaudit.alf log file to be between 524,288 bytes (512K) and 68,719,476,736 bytes (64 GB). The default size is 524,288 bytes.

The external event log (.evt file) that is generated from the cifsaudit.alf file will be larger, because the compressed contents of the cifsaudit.alf file are expanded and reformatted in the external event log file. The external event log is in Windows format. You can view it with Microsoft Event Viewer. The cifsaudit.alf log file is internally formatted and cannot be viewed with Event Viewer.

How Data ONTAP updates the event log

To save audit event information to the external event log, you can issue the `cifs audit save` command or enable automatic saving of the event information. Data ONTAP does not update the event log when the log is being viewed by a client. However, the file access information gathered when the event log is open is not lost. Data ONTAP updates the event log the next time you issue the `cifs audit save` command or an automatic save occurs.

It is important to issue the `cifs audit save` command frequently or enable frequent automatic saves to prevent loss of event information. If your event generation rate is very high, the cifsaudit.alf file will fill quickly and may overwrite older events before they are saved to the event log.

For detailed information

The following sections discuss configuring Data ONTAP to collect event information, saving and clearing event information, and understanding event log detail displays:

- "Configuring Data ONTAP for CIFS auditing" on page 163
- "Saving and clearing audit events" on page 165
- "Understanding event detail displays" on page 173

Configuring Data ONTAP for CIFS auditing

About configuring Data ONTAP for CIFS auditing

When you configure Data ONTAP for CIFS auditing, all options that are set and the event log file persist across a reboot or if CIFS is terminated or restarted.

Enabling and disabling CIFS auditing

To enable or disable CIFS auditing on the filer, complete the following step.

Step	Action
1	Enter the following command: **`options cifs.audit.enable on | off`** Use `on` to enable CIFS auditing or `off` to disable auditing. **Note** CIFS auditing is disabled by default.

Enabling and disabling auditing of file access events

To turn auditing on or off specifically for file access events, complete the following step.

Step	Action
1	Enter the following command: **`options cifs.audit.file_access_events.enable on | off`** **Note** Auditing of file access events is turned on by default. If you want this option turned on, the `cifs.audit.enable` option must also be turned on.

Enabling and disabling auditing of logon and logoff events

To turn auditing on or off specifically for logon and logoff events, complete the following step.

Step	Action
1	Enter the following command: **`options cifs.audit.logon_events.enable on | off`** **Note** Auditing of logon and logoff events is turned on by default. If you want this option turned on, the `cifs.audit.enable` option must also be turned on.

Saving and clearing audit events

About saving audit events

You can save the audit event information in the internal cifsaudit.alf file to the external event log file either manually or automatically. You can also specify a maximum size for the cifsaudit.alf file.

Saving audit events to the event log manually

To manually save audit events to the event log file, complete the following steps.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="3">1</td><td>If you want to...</td><td>Then...</td></tr>
<tr><td>Use the default event log</td><td>Go to Step 3.</td></tr>
<tr><td>Specify another file name for the event log</td><td>Go to Step 2.</td></tr>
<tr><td>2</td><td colspan="2">Enter the following command:
<tt>options cifs.audit.saveas filename</tt>
filename is the complete path name of the file to which Data ONTAP logs audit event information.
Use .evt as the file extension. Use quotes around path names that contain a space.
Examples:
<tt>options cifs.audit.saveas /etc/log/mylog.evt</tt>
<tt>options cifs.audit.saveas "/home/my event log/audit.evt"</tt></td></tr>
<tr><td>3</td><td colspan="2">Enter the following command to update the event log.
<tt>cifs audit save [-f]</tt>
Omit the <tt>-f</tt> option if the event log does not exist. Use the <tt>-f</tt> option to overwrite the existing event log.
Result: Data ONTAP writes to the event log the event information gathered since the last event log update.</td></tr>
</table>

Specifying when automatic saves occur

You can specify that audit events are automatically saved to the event log based on one or both of the following:

- The size of the internal log file, that is, how full the cifsaudit.alf file is
- A time interval

If you specify both a size threshold and a time interval, audit events will be saved to the event log whenever either the size threshold or the time interval is reached.

Enabling automatic saves based on internal log file size

To enable or disable automatic saves based on the size of the internal log file, complete the following step.

Step	Action
1	Enter the following command: **`options cifs.audit.autosave.onsize.enable on | off`**

Specifying the size threshold

If you have enabled automatic saves based on the size of the internal log file, you can specify the size threshold. The default size threshold for the internal log file is 75%, so that whenever the internal log file is 75% full, the contents are automatically saved to the external event file. You can specify the threshold as a percentage of the size of the internal log file or as an absolute size.

To specify the size threshold at which the internal log file is automatically saved, complete the following step.

Step	Action
1	Enter the following command: **`options cifs.audit.autosave.onsize.threshold Nsuffix`** *N* is the value of the size threshold. *suffix* is the unit of measure. **Example:** **`options cifs.audit.autosave.onsize.threshold 90%`** **Note** See the table below for valid values and units for the size threshold.

Valid size thresholds

The following table shows the units of measure and values you can use to specify the size threshold of the internal log file for automatic saves.

Units of measure	Values
% (percentage of the cifsaudit.alf file)	1 to 100
k (kilobytes)	1 to 67108864
m (megabytes)	1 to 65526
g (gigabytes)	1 to 64

Enabling automatic saves based on a time interval

To enable or disable automatic saves based on a time interval, complete the following step.

Step	Action
1	Enter the following command: `options cifs.audit.autosave.ontime.enable on | off`

Specifying the time interval

If you have enabled automatic saves based on a time interval, you can specify the time interval. The default time interval is 12 hours, so the contents of the internal log file are saved to the external event file every 12 hours.

To specify a different time interval for automatically saving the internal log file to the external event file, complete the following step.

Step	Action
1	Enter the following command: **`options cifs.audit.autosave.ontime.interval Nsuffix`** *N* is the value of the time interval. *suffix* is the unit of measure. **Example:** **`options cifs.audit.autosave.ontime.interval 1d`** **Note** See the table below for valid values and units for the time interval.

Valid time intervals

The following table shows the units of measure and values you can use to specify the time interval for automatic saves.

Units of measure	Values
s (seconds)	1 to 60
m (minutes)	1 to 60
h (hours)	1 to 24
d (days)	1 to 7

How automatically saved event files are named

Each time the internal log file is automatically saved to the external event file, an extension is added to the base name of the event file. You can select one of the following types of extensions to be added:

- counter
- timestamp

The filer saves the event files for up to six weeks. You can specify a limit to the number of event files that can be saved.

Counter extensions

If you select counter for automatic file naming, the extension is a number value. When an automatic save occurs, the old event files are renamed using sequentially numbered extensions. The newest event file does not have a number value added to it. For example, if the base file name is eventlog, when an automatic save occurs, the newest event file is named eventlog.evt, the previous eventlog.evt file is copied to eventlog1.evt, the eventlog1.evt file is copied to eventlog2.evt, and so on.

Specifying counter extensions

To specify counter (numbered) extensions for automatically saved event files, complete the following step.

Step	Action
1	Enter the following command: `options cifs.audit.autosave.file.extension counter`

Timestamp extensions

If you select timestamp for automatic file naming, the file name is in the following format:

base name of event file.*YYYYMMDDHHMMSS*.evt

YYYY is the 4-digit year.

MM is the 2-digit month.

DD is the 2-digit day.

HH is the 2-digit hour.

MM is the 2-digit minute.

SS is the 2-digit second.

Specifying timestamp extensions

To specify timestamp extensions for automatically saved event files, complete the following step.

Step	Action
1	Enter the following command: **`options cifs.audit.autosave.file.extension timestamp`**

Specifying the maximum number of automatically saved files

To specify the maximum number of event files that can be saved automatically, complete the following step.

Step	Action
1	Enter the following command: **`options cifs.audit.autosave.limit`** ***`value`*** *value* is a number from 1 to 999.

Specifying the maximum size of the cifsaudit.alf file

To specify the maximum size of the cifsaudit.alf file, complete the following step.

Step	Action
1	Enter the following command: **`options cifs.audit.logsize`** ***`size`*** *size* is the number of bytes. If you enter an invalid number, a message displays the range of acceptable values. **Caution** Data ONTAP overwrites the oldest data after the cifsaudit.alf file reaches the maximum size. To prevent loss of event data, you should save the cifsaudit.alf file before it is filled. By default, when the file is 75% full, a warning message is issued. Additionally, warning messages are sent when the file is nearly full and data is about to be overwritten, and when data has already been overwritten.

SNMP traps for auditing events

If you want CIFS clients to receive SNMP traps for auditing events, you must register the clients using the SNMP feature of Data ONTAP. All registered clients must have SNMP software to listen for SNMP traps.

An SNMP trap is issued whenever any of the following occurs:

- The specified time interval is reached and the cifsaudit.alf file is saved.
- The specified size threshold is reached and the cifsaudit.alf file is saved.
- The default size threshold, 75% full, is reached and the cifsaudit.alf file is in danger of wrapping and overwriting event data, but the file is not saved because the `cifs.audit.autosave.onsize.enable` and `cifs.audit.autosave.ontime.enable` options are turned off.
- The cifsaudit.alf file has wrapped and event data has been overwritten, because none of the automatic save options are turned on.

Clearing the cifsaudit.alf file

To clear the internal cifsaudit.alf file, complete the following step.

Step	Action
1	Enter the following command: **`cifs audit clear`** **Result:** If the audit has started, the internal cifsaudit.alf log file is cleared. If the audit has stopped, the cifsaudit.alf file is deleted. The external event log is not affected by this command.

Viewing the event log

To view the external event log (.evt file), complete the following steps.

Step	Action
1	Start the Event Viewer according to your Windows documentation.

Step	Action
2	In the Log menu, choose Open. **Note** Do not try to open the event log by choosing Select Computer from the Log menu and double-clicking the filer name. If you do, the Event Viewer displays the error message “The RPC server is unavailable,” because Data ONTAP does not communicate with the Event Viewer with RPC calls.
3	Choose the event log on the filer.

Understanding event detail displays

Types of event detail displays

The following event detail displays are available:

- Network logon
- Unsuccessful network logon
- Network logoff
- Windows file access
- UNIX file access
- Unsuccessful file access
- Lost record event
- Clear audit log event

Events that Data ONTAP can audit

The following table lists the events that Data ONTAP can audit.

Event ID	Event	Description	Category
516	AdtEvntDiscard	Audit events were lost	Audit Log
517	AdtLogClear	Audit log was cleared	Audit Log
529	AdtUnknownUser	Unknown user name or bad password	Logon/Logoff
530	AdtCantLogonNow	Account logon time restriction	Logon/Logoff
531	AdtAccountDisabled	Account currently disabled	Logon/Logoff
532	AdtUserAccountExpired	User account has expired	Logon/Logoff
533	AdtCantLogonHere	User can’t log on to this computer	Logon/Logoff
534	AdtLogonTypeRestricted	User not granted logon type here	Logon/Logoff
535	AdtPasswordExpired	User’s password has expired	Logon/Logoff
536	AdtNetLogonInactive	The NetLogon component is not active	Logon/Logoff
537	AdtUnsuccessfulLogon	Logon failed for reasons other than above	Logon/Logoff

Event ID	Event	Description	Category
538	AdtUserLogoff	Local or network user logoff	Logon/Logoff
539	AdtLockedOut	Account locked out	Logon/Logoff
540	AdtSuccessfulNetLogon	Network (CIFS) logon	Logon/Logoff
560	AdtObjOpen	Object (file or directory) open	File Access
563	AdtObjOpenForDelete	Object (file or directory) open for deletion	Logon/Logoff

Windows file access detail displays

Windows file access detail displays show the following types of information.

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.

Field	Description
Primary Logon ID	A unique identifier assigned when the primary user logged on.
Client 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 detail displays, but instead of an object name, NFS access appears because the file is accessed through NFS. In addition, UNIX file access detail displays show the following information about the file that you are auditing:

- 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 `find -inum` command from an NFS client.

Unsuccessful file access detail displays

Unsuccessful file access detail displays show failed attempts to 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 displays

If Data ONTAP cannot create an audit record, the lost record event detail displays give a reason, such as the following:

```
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
```

Understanding oplocks

About oplocks

Oplocks (opportunistic locks) enable 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 read from or write to a file 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 invalidate cached data and flush writes and locks. The client must then relinquish the oplock and access to the file. If there is a network failure during this flush, it can result 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 the 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.
- The CIFS clients are on an unreliable network.
- You are handling critical data and you cannot afford even the slightest data loss.

Otherwise, you can leave CIFS oplocks on.

How filer oplock setting interacts with client oplock setting

You can turn CIFS oplocks on or off at individual clients using a Windows registry setting. 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.

Managing oplocks

Setting a system-wide oplock

There is a system-wide oplock setting on the filer. If you enable this oplock setting, you can enable or disable oplocks for individual qtrees.

To set the system-wide oplock setting, complete the following step.

Step	Action
1	Enter the following command: **`options cifs.oplocks.enable [ on | off ]`** **Result:** If the `cifs.oplocks.enable` option is set to On, the oplock setting per qtree takes effect. Otherwise, the oplocks for all qtrees are disabled regardless of the per-qtree oplock setting.

Setting an oplock for each qtree

To enable or disable an oplock for an individual qtree, complete the following step.

Step	Action
1	Enter the following command: **`qtree oplocks`** ***`qtree_name`*** **`[ enable | disable ]`** *qtree_name* is the name of the qtree. **Result:** If the `cifs.oplocks.enable` option is set to On, the `qtree oplocks` command for a qtree takes effect immediately. If the `cifs.oplocks.enable` option is Off, the `qtree oplocks` command does not take effect until the option is changed to On.

About the delay time for sending oplock breaks

If a client sends a request to open a file at the same time the filer needs to send an oplock break to that client, the filer delays sending an oplock break according to the delay time value set by the command `options cifs.oplocks.opendelta`. This delay is necessary to work around a race condition that exists in some CIFS

clients. The default delay time is 8 milliseconds. This means that after the filer receives or responds to a request to open a file, the filer will make sure that 8 milliseconds have elapsed before sending an oplock break.

Changing the delay time for sending oplock breaks

You can change the default delay time for sending oplock breaks. For example, you might want to increase the delay time if you issue the `cifs stat` command and the output shows a non-zero value for the OpLkBkNoBreakAck field.

You may also see syslog messages similar to the following example and want to increase the delay time for sending oplock breaks.

Example:

```
Mon Jan 21 15:18:38 PST [CIFSAdmin:warning]: oplock break timed out
to station JOHN-PC for file \\FILER\share\subdir\file.txt
```

Changing the delay time for sending oplock breaks

To change the delay time for sending oplock breaks, complete the following step.

Step	Action
1	Enter the following command: **`options cifs.oplocks.opendelta`** ***`time`*** *time* is the delay time in milliseconds. **Note** Changing the delay time value delays the response to all file-open requests. Do not set this number higher than 35 without first consulting Network Appliance Technical Support.

Displaying CIFS session information

Types of session information

You can display the following types of session information:

- A summary of session information, which includes filer information and the number of open shares and files opened by each connected user.
- Share and file information about one connected user or all connected users, which includes
  - The names of shares opened by a specified connected user or all connected users
  - The access levels of opened files
- Security information about a specified connected user or all connected users, which includes the UNIX UID and a list of UNIX groups and Windows groups to which the user belongs.

Note

The number of open shares shown in the session information includes the hidden IPC$ share.

Different ways to specify a user

To display session information about a connected user, you can specify the user by the user name or the IP address of the workstation. In addition, if the user is connected to the filer from a pre–Windows 2000 client, you can specify the name of the workstation.

Clients sometimes connect with an unauthenticated “null” session. Such sessions are sometimes used by clients to enumerate shares. If a client has only the null session connected to the filer, you will see the following status message:

```
User (or PC) not logged in
```

Displaying a summary of session information

To display a summary of session information, complete the following step.

Step	Action
1	Enter the following command: **`cifs sessions`**

Displaying share and file information about one or all connected users

To display share and file information about connected users, complete the following steps.

Step	Action	
1	**If you want to display...**	**Then...**
	Information about one connected user	Go to Step 2.
	Information about all connected users	Go to Step 3.
2	Enter the following command: **`cifs sessions user_name | IP_address | workstation_name`**	
3	Enter the following command: **`cifs sessions *`**	

Displaying user security information

To display security information about connected users, complete the following steps.

Step	Action	
1	**If you want to display...**	**Then...**
	Information about one connected user	Go to Step 2.
	Information about all connected users	Go to Step 3.
2	Enter the following command: **`cifs sessions -s user_name | IP_address | workstation_name`**	
3	Enter the following command: **`cifs sessions -s`**	

Timing out idle sessions

About idle sessions

If a user does not have a file opened on the filer, the session is considered idle. By default, Data ONTAP disconnects a session after it has been idle for 1,800 seconds. You can specify the amount of time that elapses before Data ONTAP disconnects an idle session.

If an idle session is disconnected, it will automatically reconnect the next time the client accesses the filer.

Specifying timeout for an idle session

Complete the following step to specify the amount of idle time that elapses before Data ONTAP disconnects a session.

Step	Action
1	Enter the following command: **`options cifs.idle_timeout`** ***`time`*** *time* is the number of seconds. **Result:** The new value for this option takes effect immediately.

Disabling CIFS service

Different methods of disabling CIFS service

You can disconnect selected clients from Server Manager or the Data ONTAP command line without interrupting CIFS service for other clients.

You can disable CIFS service for the entire filer using the Data ONTAP command line. This disconnects CIFS clients for the entire filer.

Effect of disconnecting CIFS clients

When you disconnect CIFS clients that have open files on the filer, they might lose data.

How to avoid loss of client data when you disable CIFS

Always warn users before disabling CIFS so they can save their changes before closing the files. Make sure that Windows 95 and Windows for Workgroups clients have the WinPopup program configured so they can receive the alert message about a disconnection.

For detailed information

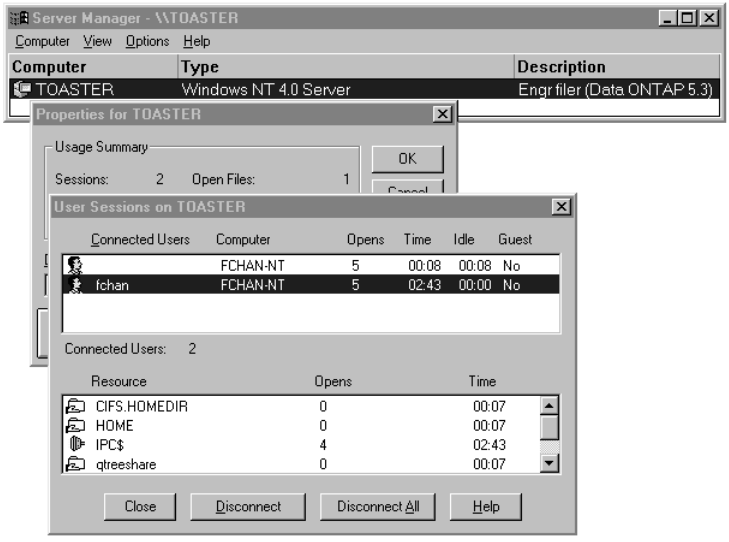
The following sections discuss how you disconnect CIFS clients and disable CIFS for the entire filer:

- "Disconnecting selected clients" on page 185
- "Disabling CIFS for the entire filer" on page 187
- "Specifying which users receive CIFS shutdown messages" on page 189

Disconnecting selected clients

Disconnecting selected clients with Server Manager

To disable CIFS service for selected clients from Server Manager, complete the following steps.

Step	Action
1	Select your filer from Server Manager.
2	Disconnect users from the filer as you would on a Windows NT server. **Example:** In the following figure, clicking Disconnect disconnects the selected client. Clicking Disconnect All disconnects all users.  **Result:** The selected clients are disconnected from the filer. However, the filer continues to run CIFS.

Disconnecting a selected user from the command line

To disconnect a CIFS client from the Data ONTAP command line, complete the following steps.

Step	Action
1	Enter the following command to display a list of connected clients: **`cifs sessions *`**
2	Enter the following command to disconnect a client: **`cifs terminate`** ***`client_name`*** **`[ [-t]`** ***`time`*** **`]`** *client_name* is the name of the workstation that you want to disconnect from the filer. *time* is the number of minutes before the client is disconnected from the filer. Entering 0 disconnects the client immediately. **Note** If you do not specify *time* and Data ONTAP detects an open file with the client, Data ONTAP prompts you for the number of minutes it should wait before it disconnects the client. **Example:** `cifs terminate jsmith-pc -t 5` **Result:** Data ONTAP sends a message to the workstation named jsmith-pc, notifying the user of the impending disconnection. Five minutes after you enter the command, jsmith-pc is disconnected from the filer.

Disabling CIFS for the entire filer

Effect of filer reboot on CIFS

The disabling of CIFS service is not persistent across reboots. If you reboot the filer after disabling CIFS service, Data ONTAP automatically restarts CIFS.

Effect of disabling CIFS on cifs commands

After you disable CIFS for the entire filer, most `cifs` commands become unavailable. The cifs commands you can enter are

- `cifs prefdc`
- `cifs restart`
- `cifs setup`
- `cifs testdc`

Disabling CIFS service for the entire filer

To disable CIFS service, complete the following step.

Step	Action
1	Enter the following command to disable CIFS service: **`cifs terminate [ -t time ]`** *time* is the number of minutes before the filer disconnects all clients and terminates CIFS service. Entering 0 makes the command take effect immediately. **Note** If you enter the `cifs terminate` command without an argument and Data ONTAP detects an open file with any client, Data ONTAP prompts you for the number of minutes it should wait before it disconnects the client. **Example:** `cifs terminate -t 5` **Result:** Data ONTAP sends a message to all connected clients, notifying the users of the impending disconnection. Five minutes after you enter the command, all clients are disconnected and the filer stops providing CIFS service.

Canceling the cifs terminate command

To cancel a `cifs terminate` command before it is executed, complete the following step.

Step	Action
1	Press Ctrl-C.

Preventing CIFS from starting upon a reboot

To prevent CIFS from starting automatically after the filer reboots, complete the following step.

Step	Action
1	Rename the /etc/cifsconfig.cfg file.

Specifying which users receive CIFS shutdown messages

Changing the setting for CIFS shutdown messages

When you issue the `cifs terminate` command, by default Data ONTAP sends a message to all open client connections to notify users when CIFS service will be disconnected. You can change the default setting so that Data ONTAP never sends these messages or sends them only to connected clients that have open files.

To specify which users receive CIFS shutdown messages, complete the following step.

Step	Action
1	Enter the following command: **`options cifs.shutdown_msg_level 0 | 1 | 2`** Use `0` to never send CIFS shutdown messages. Use `1` to send messages only to connected clients that have open files. Use `2` to send messages to all open connections, which is the default setting.

Restarting CIFS service

Restarting CIFS for the entire filer

To restart CIFS service for the entire filer, complete the following step.

Step	Action
1	Enter the following command: `cifs restart` **Result:** The filer connects to the domain controller and restarts CIFS service.

Reconfiguring CIFS for the filer

Meaning of reconfiguring CIFS

Reconfiguring CIFS means running the `cifs setup` program again for CIFS settings. The CIFS configuration settings that you can change by running `cifs setup` are as follows:

- WINS server addresses
- Whether the filer is multiprotocol or NTFS-only
- Whether the filer uses Windows domain authentication, Windows workgroup authentication, or UNIX password authentication
- Domain or workgroup to which the filer belongs
- Filer name

Prerequisites for reconfiguring CIFS

You must meet the following prerequisites before you can reconfigure CIFS:

- CIFS service must be terminated.
- If you want to change the filer's domain, the filer must be able to communicate with the primary domain controller for the domain in which you want to install the filer. You cannot use the backup domain controller for installing the filer.
- If you want to change the name of the filer, you must create a new computer account on the domain controller. This is not necessary if you are using Windows 2000.

How the cifs setup program works

The `cifs setup` program prompts you for CIFS configuration information and writes the new information to the /etc/cifsconfig.cfg file and /etc/cifssec.cfg file. The following events take place after you start the `cifs setup` program:

- The /etc/cifsconfig.cfg file is renamed to /etc/cifsconfig.bak.
- The program prompts for the WINS server addresses.
- The program prompts you to configure your filer to be NTFS-only or multiprotocol.
- If the /etc/cifsconfig.cfg file exists but is corrupted, the program prompts you to create a new computer account for the filer on the domain controller.
- If the /etc/cifsconfig.cfg file exists but is not corrupted, the program prompts you to delete the existing account. You delete the account if you are

installing the filer into a new domain. The following list describes the result, depending on your response to this prompt:

- If you delete the existing account, Data ONTAP renames /etc/cifssec.cfg to /etc/cifssec.bak. When the filer restarts, it uses the new security information specified in the new computer account on the domain controller.
- If you do not delete the existing account, the filer restarts in the same domain, using the existing security information.

- The program prompts you to install the filer in a domain or workgroup and prompts for the domain or workgroup name.
- If all information you entered is valid, the program writes the configuration information to the /etc/cifsconfig.cfg file. Otherwise, it renames the /etc/cifsconfig.bak file to /etc/cifsconfig.cfg. When the filer restarts, it uses the previous information.

Note

If `cifs setup` fails, the /etc/cifssec.bak file is not renamed to /etc/cifssec.cfg.

Effects of changing an NTFS-only filer to a multiprotocol filer

Although you can change the filer from NTFS-only to multiprotocol using `cifs setup`, you can achieve the same effects by simply setting the `wafl.default_security_style` option to `unix`.

The following list describes the effects of changing an NTFS-only filer to a multiprotocol filer:

- Existing ACLs remain unchanged.
- The security style of all volumes and qtrees remains unchanged.
- When you create a volume, its default security is `unix`.
- The `wafl.default_security_style` option is set to `unix`.

Note

Because the security style of the root volume remains `ntfs` after you change the filer to multiprotocol, you might be denied access to the root volume from UNIX as root. You can gain access if the ACL for the root volume allows full control for the Windows user that maps to root. You can also gain access by setting the `cifs.nfs_root_ignore_acl` option to `on`.

Effects of changing a multiprotocol filer to an NTFS-only filer

The following list describes the effects of changing a multiprotocol filer to an NTFS-only filer:

- If ACLs already exist on the filer root directory, /etc, and on files in the /etc directory, the ACLs remain unchanged. Otherwise, these ACLs are created such that the BUILTIN\Administrators group has full control.
- ACLs on other files and directories remain unchanged.
- The security style of all volumes, except read-only volumes, is changed to `ntfs`.
- If the /etc directory is a qtree, its security style is changed to `ntfs`.
- Security style for all other qtrees remains unchanged.
- When you create a volume or qtree, its default security style is `ntfs`.
- The `wafl.default_security_style` option is set to `ntfs`.

Effects of changing the filer's domain

After you change the filer's domain, Data ONTAP updates the membership of the BUILTIN\Administrators group to reflect the new domain. This change ensures that the new domain's Administrators group can manage the filer even if the new domain is not a trusted domain of the old domain.

Reconfiguring CIFS

To reconfigure a filer for CIFS, complete the following steps.

Step	Action
1	Enter the following command: **`cifs terminate`** **Result:** CIFS service is disabled for the filer.
2	Enter the following command: **`cifs setup`** **Result:** Data ONTAP runs the `cifs setup` program, which displays a list of prompts for you to reconfigure CIFS.

Aborting CIFS reconfiguration

To terminate the `cifs setup` program when it is in progress, complete the following step.

Step	Action
1	Press Ctrl-C. **Result:** The cifsconfig.bak file is renamed to cifsconfig.cfg. You can restart CIFS using the configuration information that was used before.

Selecting domain controllers

How the filer selects a domain controller

By default, after the filer discovers the domain controllers, it randomly selects a domain controller to use, with preference given to those on the same subnet as the filer. If you have Windows 2000 sites set up, the filer will give preference to domain controllers in the same site as the filer.

You can specify the order in which a filer contacts the domain controllers after it discovers them. The specified domain controllers are called preferred domain controllers. If the filer fails to reach any of the preferred domain controllers, it randomly selects a domain controller, with preference given to those on the local subnet or in the same Windows 2000 site.

If you want the filer to use the preferred domain controllers immediately, you can disconnect the filer from the current domain controller, which forces the filer to reestablish the connection using one of the preferred domain controllers.

The selection of preferred domain controllers is persistent across filer reboots, unless the connection fails or you reset the domain controller.

Specifying a list of preferred domain controllers

To specify a list of preferred domain controllers, complete the following step.

Step	Action
1	Enter the following command: **`cifs prefdc add`** ***`domain address`*** **`[`** ***`address`*** **`... ]`** *domain* is the domain for which you want to specify domain controllers. *address* is the IP address of the domain controller. **Example:** The following example specifies two preferred domain controllers for the lab domain: `cifs prefdc add lab 10.10.10.10 10.10.10.11`

Displaying a list of preferred domain controllers

To display a list of preferred domain controllers, complete the following step.

Step	Action
1	Enter the following command: **`cifs prefdc print [ domain ]`** *domain* is the domain for which you want to display domain controllers. If it is omitted, the command displays preferred domain controllers for all domains. **Example:** The following example displays the preferred domain controllers for the lab domain: `cifs prefdc print lab`

Deleting preferred domain controllers

To delete preferred domain controllers, complete the following step.

Step	Action
1	Enter the following command: **`cifs prefdc delete domain`** *domain* is the domain where the preferred domain controllers reside. **Example:** The following example deletes the preferred domain controllers for the lab domain: `cifs prefdc delete lab`

Reestablishing the filer connection with a domain

To disconnect the filer from the current domain controller and establish a connection between the filer and a preferred domain controller, complete the following step.

Step	Action
1	Enter the following command: **`cifs resetdc [ domain ]`** *domain* is the domain from which the filer disconnects. If it is omitted, the filer disconnects from the domain in which the filer is installed. **Example:** The following example disconnects the filer from the domain controllers for the lab domain and establishes a connection between the filer and one of the preferred domain controllers that you specified earlier: `cifs resetdc lab`

Disabling NetBIOS over TCP

About NetBIOS over TCP

NetBIOS over TCP is the standard protocol used for CIFS prior to Windows 2000. The option to use this protocol is enabled on the filer by default. It corresponds to the "Enable NetBIOS over TCP" setting in the Windows 2000 Advanced TCP/IP settings tab. If NetBIOS over TCP has been disabled in your Windows 2000 network, you can use this option to disable NetBIOS over TCP on your filer.

Requirements for disabling NetBIOS over TCP

The requirements for disabling NetBIOS over TCP are as follows:

- All clients must be running Windows 2000 or later.
- Only Windows 2000 domain controllers and virus scanners can be used.

Disabling NetBIOS over TCP

To disable NetBIOS over TCP, complete the following steps.

Step	Action	
1	**If...**	**Then...**
	CIFS is running	Enter the following command: `cifs terminate`
	CIFS is not running	Go to Step 2.
2	Enter the following command: `options cifs.netbios_over_tcp.enable off`	

File Access Using FTP 4

About this chapter

This chapter provides information about setting up your filer for FTP service and the authentication methods available for this service.

Topics in this chapter

This chapter discusses the following topics:

- "Understanding FTP service on the filer" on page 200
- "Setting up FTP service on your filer" on page 205
- "Denying access to the FTP service" on page 211
- "Viewing FTP log files" on page 212

Understanding FTP service on the filer

About the FTP service

By default, the FTP service is disabled on the filer; however, you can enable it as discussed later in this chapter.

You do not require a license to enable this service on your filer.

Components of the FTP service

The FTP service on the filer comprises the following components:

- FTP daemon
- Authentication database
- File access

The FTP daemon

The FTP daemon, `ftpd`, must be enabled to start the FTP service on the filer. You can enable `ftpd` by configuring the `ftpd.enable` option. For information about how to enable this option, see "Enabling the FTP service on your filer" on page 206.

Once enabled, `ftpd` listens on the standard FTP port 21 for FTP requests.

Authentication database for the FTP service

The database used to authenticate FTP clients depends on the authentication style you specify. You use the `ftpd.auth_style` option to specify the authentication style.The following three styles are available:

- UNIX
- NTLM
- Mixed (Default)

For each of these styles, in addition to setting up specific users in the database, you can also set up an anonymous user. For information about setting up an anonymous user, see "Anonymous FTP access options" on page 202.

UNIX: For this style, you use the /etc/passwd file in the root volume of your filer or the Network Information Service (NIS) server for authentication. You can also configure your filer to use both databases at the same time.

If you want to set up local authentication on your filer, you must use the /etc/passwd file. This file must contain the user name, password, UID, GID, and home directory information for each FTP client. The anonymous user *ftp* exists in the /etc/passwd file by default; however, to enable anonymous access to your filer, you must configure the options discussed in "Anonymous FTP access options" on page 202.

If you use NIS authentication, you must enable the `nis.enable` option on your filer and have the users configured appropriately on the NIS server.

If you configure your filer to use both databases at the same time, the order in which these databases get used is determined by the passwd map in the /etc/nsswitch.conf file. This file specifies a default order in which these databases are used; however, you can edit the file. For information on how to edit the /etc/nsswitch.conf file, see the *Network Management Guide*.

NTLM: For this style, the authentication goes through the Windows Domain Controller (DC). This style offers a more secure method of authentication because the user name and password are encrypted.

The format for the user name can be one of the following:

- Domain\username
- Username@domain
- Username (If no domain is specified, by default, the domain that the filer is currently in is used.)

The home directory for each user is specified by a combination of the `cifs.home_dir` option and the user name, that is, the home directory is of the format cifs.home_dir\username.

Note

The path specified by the `cifs.home_dir` option is case-sensitive. The user name is not case-sensitive. For example, if cifs.home_dir is \home and user name is JOHN, the home directory for the user is \home\john.

You can also set up anonymous (guest) access to your FTP server when using NTLM for authentication. For more information about anonymous access, see "Configuring anonymous access to your filer" on page 210.

Mixed: This style is used for authentication by default. The database used to authenticate an FTP client depends on the format of the user name specified at login time. If the user name is specified without a domain name, UNIX

authentication style is used. If a domain name is specified with the user name in either domain\username or username@domain format, NTLM authentication style is used.

For example, if an FTP client specifies the user name as joe@doe, the NTLM authentication will be used; however, if the client specifies the user name joe, UNIX authentication will be used.

Anonymous FTP access options

By default, anonymous FTP access to the filer is disabled; however, you can enable it. The following table lists the options in Data ONTAP for setting up anonymous access to the filer.

Option	Description	Values (Default)
`ftpd.anonymous.enable`	Enables or disables `ftpd` daemon.	On | Off (Off)
`ftpd.anonymous.name`	Enables you to specify a different name from the default, *anonymous*, for anonymous login.	*name_string* (anonymous)
`ftpd.anonymous.home_dir`	Specifies the directory to which the anonymous user will be mapped.	*pathname* (/vol/vol0/home/ftp) **Note** The directory *ftp* specified above is not created automatically; you must manually create it. If you change the value of this option to something other than default, you must create a directory corresponding to that value.

For the UNIX authentication style, although the /etc/passwd file contains a special entry, *ftp*, the users must use the value specified by the `ftpd.anonymous.name` option for anonymous access. The home directory specified by the *ftp* entry in the /etc/passwd file is overridden by the `ftpd.anonymous.home_dir` option.

For the NTLM authentication style, the access privileges of a NULL user apply for anonymous access.

For the mixed authentication style, the authentication style used for anonymous access depends on the format of the user name specified.

File access permissions

The authentication mechanism used for FTP access only identifies an FTP client to the FTP server; it does not enable the client to access files in a qtree. The client must have appropriate qtree security permissions to access the files in a qtree. For more information about qtree security permissions, see the chapter about data organization using volumes and qtrees in the *Storage Management Guide*.

Prohibiting access to the FTP service

If you want to prohibit users from using the FTP service on your filer, you can include their user names in the /etc/ftpusers file. This file is not created automatically; you must create it in the /etc directory of your filer. This file can be used for any authentication style. For more information about how to create this file and the information to include, see "Denying access to the FTP service" on page 211.

Other configurable FTP service options

The following table lists the other configurable options available for the FTP service.

Option	Description	Value (Default)
`ftpd.idle_timeout`	Specifies the time in seconds, minutes, or hours after which an idle connection is disconnected.	*number* {s | m | h} (900s) ◆ Valid values are between 300s and 12h ◆ Setting the value to 0 disables this option, that is, there will be no timeout on idle connections.

Option	Description	Value (Default)
`ftpd.max_connections`	Specifies the maximum number of concurrent FTP connections allowed.	*number* (1000)
`ftpd.tcp_window_size`	Specifies the TCP window size. The TCP window size specifies the amount of outstanding unacknowledged data a sender can send on a connection. **Caution** Setting this option inappropriately will adversely impact the FTP service on your filer. If you are unsure of the value to set, use the default value.	*number* in bytes (8760)

Setting up FTP service on your filer

About setting up the FTP service

Setting up the FTP service on your filer involves configuring a mandatory option, `ftpd.enable`, and a few optional options, such as `ftpd.auth_ style` (to set up the authentication style), `ftpd.anonymous.enable` (to enable anonymous access), and so on. Once the FTP service is enabled on a filer, the optional options you have not set are automatically assigned default values.

For detailed information

For detailed information about how to set up the components of the FTP service on your filer, see the following topics:

- "Enabling the FTP service on your filer" on page 206
- "Configuring an authentication style" on page 207
- "Configuring anonymous access to your filer" on page 210

Enabling the FTP service on your filer

About enabling the FTP service

You can enable the FTP service on your filer by configuring the `ftpd.enable` option.

Once you enable the FTP service, the maximum number of FTP connections your filer will allow (`ftpd.max_connections`) and the timeout value for an idle connection (`ftpd.idle_timeout`) are set to default values; however, you can change both of these options with the `options` command.

Enabling the FTP service

To enable the FTP service on your filer, complete the following step.

Step	Action
1	Enter the following command: **`options ftpd.enable on`**

Configuring an authentication style

About configuring an authentication style

You can configure one of the following authentication styles for the FTP service on your filer:

- UNIX
- NTLM
- Mixed

By default, the authentication style is set to Mixed when the FTP service is enabled.

For more information about these authentication styles, see "Authentication database for the FTP service" on page 200.

Limitation of NTLM authentication style

Data ONTAP allows only lowercase names for user home directories. Even if you specify the user name in uppercase, you must make sure that a lowercase name is specified for the user's home directory; otherwise, the user will not be able to access files after logging in.

Configuring the UNIX authentication style

To configure the UNIX authentication style for the FTP service on your filer, complete the following step.

Step	Action	
1	**If...**	**Then...**
	You want to use only the /etc/passwd file for authentication and have NIS disabled on your filer	Enter the following command: **`options ftpd.auth_style unix`**
	You want to use only the /etc/passwd file for authentication and have NIS enabled on your filer	Enter the following command: **`options ftpd.auth_style unix`**
		In the /etc/nsswitch.conf file, edit the passwd map as follows: **`passwd: files`**
	You want to use only NIS for authentication	Enter the following commands: **`options ftpd.auth_style unix`** **`options nis.enable on`**
		In the /etc/nsswitch.conf file, edit the passwd map as follows: **`passwd: nis`**
	You want to use both /etc/passwd and NIS for authentication	Enter the following commands: **`options ftpd.auth_style unix`** **`options nis.enable on`**
		In the /etc/nsswitch.conf file, edit the passwd map as follows: **`passwd: files nis`**

Configuring the NTLM authentication style

To configure the NTLM authentication style for the FTP service on your filer, complete the following step.

Step	Action
1	Enter the following command: **`options ftpd.auth_style ntlm`** **`options cifs.home_dir`** ***`pathname`*** where *pathname* is location of user home directories; for example, \home. **Note** The home directory of a user is a combination of *pathname* specified by the `cifs.home_dir` option and the user ID of the user. The path name specified by the `cifs.home_dir` option is case-sensitive; however, the user ID is not case-sensitive. For example, if `cifs.home_dir` is \home and the user name is JOHN, the home directory for the user is \home\john.

Configuring the Mixed authentication style

To configure the Mixed authentication style on your filer, complete the following step.

Note

If you want to use NIS for authentication of users specifying their user name without a domain name, make sure that the `nis.enable` option on your filer is set to On. For more information, see "Authentication database for the FTP service" on page 200 and "Configuring the UNIX authentication style" on page 208.

Step	Action
1	Enter the following command: **`options ftpd.auth_style mixed`**

Configuring anonymous access to your filer

About options to configure anonymous access

Anonymous access to your filer is disabled by default. You must use the `ftpd.anonymous.enable` option to enable it. In addition, you can configure the following options:

- `ftpd.anonymous.name`
- `ftpd.anonymous.home_dir`

Note

If the authentication style configured for the FTP service on your filer is NTLM, the `ftpd.anonymous.home_dir` option must be configured.

For more information about these options, see "Anonymous FTP access options" on page 202.

Configuring anonymous access

To enable anonymous access to your filer, complete the following step.

Step	Action
1	Enter the following command: **`options ftpd.anonymous.enable on`**

Denying access to the FTP service

About denying access

You can configure your filer to deny the FTP service to specific users by including their user names in the /etc/ftpusers file. This file is not created automatically when the Data ONTAP software is installed on your filer; you must create it.

Format of /etc/ftpusers

The /etc/ftpusers file must contain one user name per line. For NTLM authentication, the user name must also include the domain name in one of the following formats:

- Domain\username
- Username@domain

Note

You must specify the exact domain name in the preceding formats; otherwise, access to the specified user may not be denied. For example, if you want to deny a user in NT-domain.com, you must specify the domain as NT-domain.com. If you specify NT-domain instead of NT-domain.com, the user will not be denied access.

Editing the /etc/ftpusers file

To edit the /etc/ftpusers file on your filer, complete the following steps. For more information about accessing the /etc/ftpusers file, see the section on editing configuration files in the introduction to filer administration in the *Storage Management Guide*.

Step	Action
1	Open the /etc/ftpusers file with an editor.
2	Add the user names of the users (one name per line) to whom you want to deny access.
3	Save the /etc/ftpusers file.

Viewing FTP log files

Types of log files

The ftp daemon maintains two log files:

- The ftp.cmd file stores all commands that the ftp daemon receives.
- The ftp.xfer file stores a list of all files that are transferred between the FTP clients and the filer using the FTP protocol.

Where the log files are stored

The FTP log files are stored in the /etc/log directory on the filer's default volume (/vol/vol0 by default).

Viewing log files

Data ONTAP does not contain a log file viewer. To view the log files, complete the following steps.

Step	Action
1	Access the /etc/log directory on the filer's default volume (/vol/vol0 by default) from an NFS client or a CIFS client.
2	Use a text viewer or text editor to open and view a log file.
3	Close the log file when you are finished viewing it.

File Access Using HTTP 5

About this chapter

This chapter describes how you can add HTTP service to your filer and configure the HTTP service to match your environment.

Note

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 man pages and to use FilerView.

Topics in this chapter

This chapter discusses the following topics:

- "Starting the HTTP service" on page 214
- "Testing the HTTP service" on page 215
- "Specifying the filer response to HTTP requests" on page 216
- "Configuring MIME Content-Type values" on page 224
- "Maintaining HTTP service security" on page 226
- "Using virtual hosting" on page 236
- "Displaying HTTP statistics" on page 238
- "Viewing HTTP connection information" on page 242

Starting the HTTP service

About HTTP service

You can make the filer an HTTP server by adding HTTP service to it. After you start the HTTP service, clients can display files in the directory designated as root for the HTTP service by using a Web browser.

Starting HTTP service on the filer

To start HTTP service on your filer, complete the following steps.

Note

If you want the HTTP service to remain active after rebooting, you must add them to the /etc/rc file.

Step	Action
1	Enable the `httpd` daemon by entering the following command: **`options httpd.enable on`**
2	Enter the following command to specify the root directory that contains the files and directories to be read by HTTP clients: **`options httpd.rootdir`** ***`directory`*** *directory* is the root directory name. **Example:** If the root directory is /vol0/home/users/pages, you use the following command: `options httpd.rootdir /vol0/home/users/pages`
3	If you want to change the maximum size of the /etc/log/httpd.log file from the default of 2,147,483,647 bytes (2 GB minus 1 byte), enter the following command: **`options httpd.log.max_file_size`** ***`bytes`***
4	Make a copy of the /etc/httpd.mimetypes.sample file and name the copy /etc/httpd.mimetypes. If the /etc/httpd.mimetypes file is missing, the HTTP client uses the information in the /etc/httpd.mimetypes.sample file.

Testing the HTTP service

About testing the HTTP service

You can test the HTTP service to ensure that it is working correctly. You test the service by accessing a sample file you create in the directory designated as root for the HTTP service. If you are able to access the file by using the appropriate URL in a Web browser, you have successfully enabled the HTTP service.

If you specify a directory name instead of the file name in the URL, the filer looks for the following files in the specified directory in the following order:

- index.html
- default.htm
- index.htm
- default.html

If these files do not exist, the filer generates an HTML version of the directory listing for that directory.

Testing the service

To test the HTTP service, complete the following steps.

Step	Action
1	Create an HTML file in the root directory for HTTP. **Example:** If the HTML root directory is /vol/vol0/home/users/pages, create a file named myfile.html in the root directory.
2	Start a Web browser on a client and specify the URL of the HTML file in the browser. **Example:** If your filer is toaster and the root directory for HTTP is /vol/vol0/home/users/pages, you use this URL: `http://toaster/myfile.html` The path component of the URL is a path name relative to the HTTP root. Do not specify the complete path name to the file in the URL. If you have set up the HTTP service correctly, you will see the contents of myfile.html.

Specifying the filer response to HTTP requests

About specifying the response

You can specify that the response a filer has 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. You specify the response by adding the map, redirect, pass, or fail translation rule to the /etc/httpd.translations configuration file.

How the filer processes URL translation rules

The filer processes the rules defined in the /etc/httpd.translations file in the order they are listed and applies the rule if the URL matches the template. After the first successful match, the filer stops processing the remaining rules.

For detailed information

The following sections discuss the translation rules:

- "Adding the map rule" on page 217
- "Adding the redirect rule" on page 219
- "Adding the pass rule" on page 221
- "Adding the fail rule" on page 223

Adding the map rule

What the map rule specifies

The map rule specifies that if a component of a URL matches the template, the request is mapped to another directory on the same host, as defined in the *result* field.

Adding the map rule

To add a map rule in the /etc/httpd.translations file, complete the following steps.

Step	Action
1	Open the /etc/httpd.translations file using a text editor.
2	Enter the following command: **`map template result`** *template* is a component of a URL, for example, /image-bin/graphics/ *result* is the directory to which the request is mapped.
3	Save the file.

About wildcard characters in the template field

You can use the wildcard character asterisks (*) in the *template* field. If the wildcard character matches zero or more characters, including the slash (/) character.

About wildcard characters in the result field

If you use asterisks as wildcard character in the *result* field, the wildcard character represents the text expanded from the match in the *template* field. You should include the wildcard character in the *result* field only if you use a wildcard in the template field.

Using multiple wildcard characters

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.

Example of a map rule entry

The following map rule entry in the /etc/httpd.translations file 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/*
```

Adding the redirect rule

What the redirect rule specifies

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.

Adding the redirect rule

To add a redirect rule in the /etc/httpd.translations file, complete the following steps.

Step	Action
1	Open the /etc/httpd.translations file using a text editor.
2	Enter the following command: **`redirect template result`** *template* is a component of a URL. *result* is the URL to which the request is redirected. **Note** The *result* field for the redirect rule must be specified as a complete URL beginning with http:// and the host name.
3	Save the file.

About wildcard characters in the template field

If you use asterisks (*) as wildcard characters in the *template* field, the wildcard character matches zero or more characters, including the slash (/) character.

About wildcard characters in the result field

If you use asterisks as wildcard characters in the *result* field, the wildcard character represents the text expanded from the match in the *template* field. You should include the wildcard character in the *result* field only if you use a wildcard in the *template* field.

Using multiple wildcard characters

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.

Example of a redirect rule entry

The following redirect rule entry in the /etc/httpd.translations file causes the filer to redirect Common Gateway Interface (CGI) requests to another HTTP server, named cgi-host. This is essential for calls to the /cgi-bin directory because the filer does not execute CGI requests.

```
redirect /cgi-bin/* http://cgi-host/*
```

Adding the pass rule

What the pass rule specifies

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.

Adding the pass rule

To add a pass rule in the /etc/httpd.translations file, complete the following steps.

Step	Action
1	Open the /etc/httpd.translations file using a text editor.
2	Enter the following command: **`pass template [result]`** *template* is a component of a URL. *result* is the URL to which the request is redirected. The *result* field is optional.
3	Save the file.

About wildcard characters in the template field

If you use asterisks (*) as wildcard characters in the *template* field, the wildcard character matches zero or more characters, including the slash (/) character.

About wildcard characters in the result field

If you use asterisks as wildcard characters in the *result* field, the wildcard character represents the text expanded from the match in the *template* field. You should include the wildcard character in the *result* field only if you use a wildcard in the *template* field.

Using multiple wildcard characters

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.

Example of a pass rule entry

The following pass rule entry in the /etc/httpd.translations file causes the filer to process the request for any URL containing the /image-bin directory as is:

```
pass /image-bin/*
```

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.

Adding the fail rule

What the fail rule specifies

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.

Adding the fail rule

To add a fail rule in the /etc/httpd.translations file, complete the following steps.

Step	Action
1	Open the /etc/httpd.translations file using a text editor.
2	Enter the following command: **`fail template`** *template* is a component of a URL.
3	Save the file.

Example of a fail rule entry

The following fail rule entry in the /etc/httpd.translations file causes the filer to deny access to the /user/forbidden directory:

```
fail /usr/forbidden/*
```

Configuring MIME Content-Type values

About MIME Content-Type values

The MIME (Multipurpose Internet Mail Extensions) Content-Type value of a file tells a browser on a client how to interpret the file. For example, if the MIME Content-Type value shows that a file is an image file and the client is configured properly, the browser can render the image by using a graphics program.

Note

For more information about MIME, see RFC 1521.

How the filer sends MIME Content-Type values

You can configure the filer to send the appropriate MIME Content-Type value in each response to a get request from a client. You configure the filer by mapping the file name suffix, for example, .gif, .html, or .mpg, according to information in the /etc/httpd.mimetypes file.

Mapping file name suffixes

You edit the /etc/httpd.mimetypes file to map a file name suffix to a MIME Content-Type value.

To edit the /etc/httpd.mimetypes file, complete the following step.

Step	Action
1	Edit the entries in the /etc/httpd.mimetypes file. Entries are in the following format: `# An optional comment.` *suffix Content-Type* Lines preceded by the # sign are comments. The file name suffix is not case-sensitive.

Example

The 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 end with *.pct* or *.pict* 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.

Maintaining HTTP service security

Levels of HTTP service security

You can maintain three levels of security for HTTP services:

- Use the HTTP options to restrict access.

  The HTTP options restrict access to HTTP services from specified hosts and from specified interfaces.
- Use an HTTP virtual firewall.

  An HTTP virtual firewall restricts HTTP access at the network interface level.
- Protect Web pages with user authentication.

  User authentication provides security at the directory tree level by requiring a valid user name and password before granting access.

For detailed information

The following sections discuss the levels of security for HTTP services:

- "Using HTTP options to restrict access" on page 227
- "Using an HTTP virtual firewall" on page 228
- "Protecting Web pages" on page 229
- "Editing the /etc/httpd.access file" on page 231
- "Creating and editing httpd.passwd and httpd.group files" on page 234

Using HTTP options to restrict access

Available options

The following options to restrict HTTP access are available:

- `httpd.access`—Restricts access to the HTTP services.
- `httpd.admin.access`—Restricts access to filer administration via HTTP (FilerView).

  If the `httpd.admin.access` option is set, the `trusted.hosts` option is ignored for HTTP administration.

You can restrict access on a host or a network interface basis. For more information about these options, see the na_protocolaccess(8) man page.

Example of restricting access to http services

In the following example, only host Host1 is allowed access through interface e3 to the HTTP services on filer Filer1.

```
Filer1> options httpd.access host=Host1 AND if=e3
```

Example of restricting access to httpd

In the following example, host Host1 is denied FilerView access to the filer Filer1.

```
Filer1> options httpd.admin.access host!=Host1
```

Using an HTTP virtual firewall

How the firewall provides security

An HTTP virtual firewall provides security on your filer by restricting HTTP access through the subnet interface over which the HTTP requests arrive. You restrict HTTP access by marking the subnet interface as untrusted. An untrusted subnet interface provides only read-only HTTP access to the filer. By default, a subnet interface is trusted.

When to mark a subnet interface as untrusted

Mark a subnet interface as untrusted if it meets all the following conditions:

- You know you are going to service HTTP requests over that interface.
- You do not want to allow requests through protocols other than HTTP.
- You want to restrict access to the filer through that interface to read-only access.

Restricting HTTP access

To restrict HTTP access over a subnet interface, complete the following step.

Step	Action
1	Enter the following command: **`ifconfig interface_name [trusted|untrusted]`** *interface_name* is the specific interface to set as trusted or untrusted. Use `untrusted` to restrict HTTP access or `trusted` to allow full HTTP access. **Example:** This example marks the f0 interface as untrusted. **`ifconfig f0 untrusted`**

Protecting Web pages

About Web page protection

You can restrict HTTP access, and thereby protect Web pages, by preventing unauthorized users from accessing Web pages. In this way, only specified users or groups can access directories containing the Web pages.

Data ONTAP provides the following two methods of authentication for HTTP access:

- Basic
- NTLM

You specify the method of authentication to use in the /etc/httpd.access file. Both authentication methods can coexist on a filer, but you can specify only one authentication method per directory in the HTTP subtree.

Basic authentication

You use the following three configuration files to set up the basic authentication for the HTTP service:

- /etc/httpd.access
- /etc/httpd.passwd
- /etc/httpd.group

The /etc/httpd.access file contains the method of authentication, the directories for which you want to restrict access, and the list of users and groups authorized to access these directories.

The /etc/httpd.passwd file contains the encrypted form of the password that a user, specified in the /etc/httpd.access file, uses to gain access to the directories specified in the /etc/httpd.access file. The /etc/httpd.passwd file uses the same format that the /etc/passwd file uses.

The /etc/httpd.group file contains group names and user IDs of the members of each group who are authorized to access the directories specified in the /etc/httpd.access file. The /etc/httpd.group file uses the same format that the /etc/group file uses.

NTLM authentication

You can use Windows Domain Authentication instead of basic authentication for a directory. Data ONTAP uses the Domain Controller (DC) to authenticate users accessing the directories containing the Web pages.

You must specify the directories in the /etc/httpd.access file for which you want the DC to authenticate users.

A user accessing a directory for which NTLM authentication has been set up must specify a domain with the user name. If a domain is not specified, the domain of the filer is assumed as a default. The users can specify the domain in either of the following formats:

- user_name@domain_name
- domain_name\user_name

Requirement for using NTLM authentication: You must have CIFS running on your filer to use the NTLM authentication for HTTP access.

Advantages of using NTLM authentication: First, if you use Internet Explorer™ (IE) as your browser, NTLM authentication is a more secure method of authenticating users because user name and password are not sent in clear text over the wire. However, for the Netscape™ browser, user name and password are sent from the browser to the filer in clear text, providing no additional security.

Second, you do not need to maintain information in the /etc/http.passwd and /etc/http.group files, thus centralizing user administration.

Editing the /etc/httpd.access file

About 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—Specifies the directory you want to protect. The directory directive encloses all other directives.
- AuthName—Specifies an alias for the directory that appears instead of the directory name in the browser password dialog box when a user tries to access the directory.
- require user—Specifies the users who can access the directory.
- require group—Specifies the groups that can access the directory.

Note

The directives *require user* and *require group* are only required for Basic authentication.

Format for directive information

Directive information for each directory in the /etc/httpd.access file is given in the following format:

```
<Directory directory>
directive ...
</Directory>
```

directory is the specific directory tree name for which you want to enable authorized access.

Editing the /etc/httpd.access file

To edit the /etc/httpd.access file, complete the following steps.

Step	Action
1	Open the /etc/httpd.access file for editing.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>2</td><td colspan="2">Specify the directory tree you want to protect in the following line:
<code><Directory <i>directory</i>></code>
<i>directory</i> is the specific directory tree name you want protected.</td></tr>
<tr><td rowspan="3"></td><td>If...</td><td>Then...</td></tr>
<tr><td>You are configuring authentication using /etc/httpd.passwd and /etc/httpd.group files</td><td>Specify the alias for the directory in the following line:
<code>AuthName title_phrase</code>
title_phrase is any string you specify that appears instead of the directory name in the browser password dialog box when a user tries to access the directory; this name can contain spaces.
Example: <code>AuthName Secured Area</code></td></tr>
<tr><td>You are configuring NTLM authentication</td><td>Specify the following:
<code>AuthName Windows(tm) Authentication</code>
Note
The above line should appear in the /etc/httpd.access file exactly as shown.

Go to Step 5.</td></tr>
<tr><td>3</td><td colspan="2">Specify the users who can access the directory in the following line:
<code>require user user_id [, user_id, ...]</code>
user_id is the specific user ID for each user who should have access to the directory.</td></tr>
<tr><td>4</td><td colspan="2">Specify the groups that can access the directory in the following line:
<code>require group group_id [, group_id, ...]</code>
group_id is the specific group ID for each group that should have access to the directory.</td></tr>
</table>

Step	Action
5	End the list of directives for the specified directory using the following line: `</Directory>`
6	Save the file.

Example

The following example shows the use of multiple Directory directives in a /etc/httpd.access file to use either Basic or NTLM authentication on a filer.

```
<Directory /vol/vol0/web1>
AuthName Windows(tm) Authentication
</Directory>
<Directory /vol/vol0/web2>
AuthName Web2 directory
require user test1
require group testg1
</Directory>
<Directory /vol/vol0/web3>
AuthName Windows(tm) Authentication
</Directory>
<Directory /vol/vol0/web4>
AuthName Web4 directory
require user test2
</Directory>
```

In this example, web1 and web3 use NTLM authentication and web2 and web4 use Basic authentication. Access to web2 is limited to user test1 and members of group testg1, and access to web4 is limited to user test2.

Creating and editing httpd.passwd and httpd.group files

About httpd.passwd and httpd.group files

The /etc/httpd.passwd file contains encrypted passwords of users listed in the /etc/httpd.access file. The /etc/httpd.group file contains the group names and the users belonging to those groups.

These files are only required if you are using Basic authentication to authenticate users.

Ways to create the /etc/httpd.passwd file

If you have an HTTP server that uses a user name-password method to authenticate users, copy user IDs and encrypted passwords from it. You must edit the /etc/httpd.passwd file to remove users that you do not want to have access.

If an HTTP server is not available, copy an existing /etc/passwd file from a UNIX server and save it on the filer as the /etc/httpd.passwd file.

Editing the /etc/httpd.passwd file

To edit the /etc/httpd.passwd file, complete the following steps.

Step	Action
1	Open the /etc/httpd.passwd file.
2	Remove the user IDs and encrypted passwords of users that you do not want to have access to the directory you specified in the /etc/httpd.access file.
3	Save the edits.

Ways to create the /etc/httpd.group file

If you have an HTTP server that authenticates groups of users, copy the group names and user IDs from it. You must edit the /etc/httpd.group file to remove groups that you do not want to have access.

If an HTTP server is not available, copy an existing /etc/group file from a UNIX server and save it on the filer as the /etc/httpd.group file.

Editing the /etc/httpd.group file

To edit the /etc/httpd.group file, complete the following steps.

Step	Action
1	Open the /etc/httpd.group file to edit the following line: `group_id:user_id [, user_id, ...]` The lists are copied in from a server that has a similar list.
2	Add or remove groups and users. Group and user information is listed in the following format: `group_id: user_id [user_id ...]` *group_id* is the group name. *user_id* is the name of each user who belongs to the group.
3	Save the file.

Using virtual hosting

About virtual hosting

Virtual hosting enables you to configure your filer to host several IP addresses using only one physical interface. Virtual hosting is useful if, for example, an Internet provider wants to host several Web sites but has only one physical interface.

When virtual hosting is enabled, an HTTP server uses the IP address pointed to by an incoming HTTP request to find the directory that contains the HTTP pages belonging to the virtual host.

How to enable virtual hosting

To enable virtual hosting, you

- Set up virtual hosting by editing the /etc/httpd.hostprefixes file
- Map virtual host addresses to the vh interface with the `ifconfig` command

  For more information about the vh interface, see the na_ifconfig(1) man page.

Setting up virtual hosting

You set up virtual hosting by putting subdirectory and host or address entries in the /etc/httpd.hostprefixes file. Data ONTAP redirects HTTP requests to the subdirectory and host or address listed in the /etc/httpd.hostprefixes file.

To set up virtual hosting, complete the following steps.

Step	Action
1	Open the /etc/httpd.hostprefixes file for editing.
2	Enter subdirectory and host or address entries in the following format: ***prefix [host-name-or-address ...]*** *prefix* is a subdirectory in the HTTP root directory, which is defined by the `options httpd.rootdir` command. *host-name-or-address* is an HTTP host name or an IP address. You can have more than one of each.
3	Save the edits.

Example

Assume that you added the following entry in the /etc/httpd.hostprefixes file:

```
/customer 192.225.37.102
/customer www.customer.com
```

If the HTTP server receives an HTTP request destined for one of its virtual host IP addresses, 192.225.37.102, the IP address is used to select the virtual host root directory, the /customer directory, from the /etc/httpd.hostprefixes file. An HTTP request with an HTTP 1.1 Host: header specifying *www.customer.com* is directed to the /customer directory. In either case, the requestor cannot get a file outside the /customer directory.

Mapping virtual host addresses

You must map virtual host addresses to the vh interface to support virtual hosting. To map virtual host addresses to the vh interface, complete the following step.

Step	Action
1	Enter the following command to add a new IP virtual host address mapping: **`ifconfig vh -wins [alias | -alias]` *`address`*** Use `alias` to map a virtual host address. Use `-alias` to remove a virtual host address map. *address* is the IP address you want to add or remove. **Note** If you need to create a virtual subnet with many contiguous addresses, the IP address can be a subnet address. The `vh` interface indicates that you are adding a virtual host address rather than adding an IP alias address to a network interface.

Displaying HTTP statistics

Types of HTTP statistics displayed

You can use the `httpstat` command to display the following five types of statistics about HTTP operations on the filer:

- Request
- Detailed
- Error
- Service
- Timeout

Request statistics

If you specify request statistics, Data ONTAP displays the following statistics.

Column	Description
`Accept`	Number of new connections accepted by the filer
`Reuse`	Number of new requests received on existing connections
`Response`	Number of responses sent
`InBytes`	Number of bytes received for all incoming requests
`OutBytes`	Number of bytes sent, including all HTTP headers, but not including data generated by servlets

Detailed statistics

If you specify detailed statistics, Data ONTAP displays the following statistics.

Column	Description
`Get`	Number of requests for files received
`Head`	Number of requests for file information received
`Redirect`	Number of requests redirected to another file

Column	Description
NotMod	Number of times clients (browsers) are told that requested files are not modified
Post	Number of POST requests received
Put	Number of PUT requests received
Servlet	Number of servlet requests received

Error statistics

If you specify error statistics, Data ONTAP displays the following statistics.

Column	Description
Errors	Number of HTTP protocol error responses returned
BadReq	Number of unrecognized requests received
LogDiscard	Number of log entries discarded because the log was full
UnAuth	Number of requests denied because they lacked authorization
RcvErr	Number of requests aborted because of errors on the input socket

Service statistics

If you specify service statistics, Data ONTAP displays the following statistics.

Column	Description
Open	Number of currently open connections
Peak	Maximum number of connections ever achieved
Waits	Current number of connections accepted, but waiting for a connection structure

Timeout statistics

If you specify timeout statistics, Data ONTAP displays the following statistics.

Column	Description
`Pending`	Number of connection structures reclaimed after the network connection was started, but before any data was sent to the filer
`Active`	Number of connection structures reclaimed after the network connection was started and a partial request was sent, but before the complete request arrived
`Idle`	Number of connections that were reclaimed after a complete request, but before the open connection could receive another request

Displaying HTTP statistics

To display HTTP statistics, complete the following step.

Step	Action
1	Enter the following command: **`httpstat [-dersta]`** The `-d` option displays detailed statistics. The `-e` option displays error statistics. The `-r` option displays request statistics. The `-s` option displays service statistics. The `-t` option displays timeout statistics. The `-a` option displays all HTTP statistics. If you use no arguments, `httpstat` displays HTTP request statistics. For detailed information about the `httpstat` command, see the na_httpstat(1) man page.

Resetting HTTP statistics

To reset HTTP statistics, complete the following step.

Step	Action
1	Enter the following command: **`httpstat [-zderta]`** The `-zd` option resets detailed statistics. The `-ze` option resets error statistics. The `-zr` option resets request statistics. The `-zt` option resets timeout statistics. The `-za` option resets all HTTP statistics except the service statistics. **Note** You cannot reset the service statistics. For detailed information about the `httpstat` command, see the na_httpstat(1) man page.

Viewing HTTP connection information

Types of HTTP connection information

The following types of information for each HTTP connection are listed in the /etc/log/httpd.log file:

- IP address of HTTP client.
- Names of authorized users making requests. If the page is protected, Data ONTAP lists authorized names it gets from 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; Greenwich Mean Time (GMT) is used.
- Request line from connecting host, for example, `get /netapp.html`.
- Status code returned by the server, as defined in the HTTP 1.0 specifications.
- Total bytes sent in response by the filer, not including the MIME header.

Viewing the /etc/log/httpd.log file

To view the /etc/log/httpd.log file, complete the following step.

Step	Action
1	Access the /etc/log directory on the filer default volume (/vol/vol0 by default) from an NFS or CIFS client.
2	Use a text viewer or text editor to open and view the httpd.log file.
3	Close the log file when you are finished viewing it.

Example

The 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
```

Changing the /etc/log/httpd.log file format

The default format of the /etc/log/httpd.log file shows the IP address of the HTTP clients and the HTTP path accessed, but not which virtual host is accessed.

To change the format of the /etc/log/httpd.log file so that it distinguishes HTTP messages by virtual hosts, complete the following step.

Step	Action
1	Enter the following command: `options httpd.log.format alt1`

File Access Using DAFS 6

About this chapter

This chapter provides the information you need to manage the Direct Access File System (DAFS) protocol. When DAFS is configured and enabled, file access is transparent. However, you might need to manage exports, measure performance, and display status for the DAFS service.

Topics in this chapter

This chapter discusses the following topics:

- "Understanding the DAFS protocol" on page 246
- "Understanding the DAFS service implementation" on page 248
- "Configuring the DAFS service" on page 249
- "Starting and stopping the DAFS service" on page 253
- "Attaching and detaching network devices" on page 257
- "Understanding session authentication" on page 260
- "Managing exports for DAFS service" on page 265
- "Managing client host access to DAFS exports" on page 270
- "Measuring DAFS performance" on page 275
- "Displaying DAFS status" on page 280
- "Configuring DAFS with options" on page 289

Understanding the DAFS protocol

DAFS protocol overview

The DAFS protocol defines a set of file access and management operations similar to Network File System (NFSv4), but optimized for the local file-sharing and clustered environments. A local file-sharing environment is one that is geographically constrained (one or a few machine rooms) with mostly homogeneous clients. It is managed from a single administrative control point and focused on supporting high-performance applications. An example is a data center.

How DAFS optimizes I/O performance

The DAFS protocol provides high-performance file input/output (I/O), improved CPU utilization, and low networking overhead by transporting data over a direct access transport (DAT) interface. DAT is an interface to memory-to-memory networks.

Note

This version of Data ONTAP supports virtual interface over Gigabit Ethernet (VI/IP) memory-to-memory networking only.

The DAFS protocol optimizes application throughput with streamlined messaging composition and transport. Flow control enables multiple application servers to access a single NetApp filer without congestion.

About direct access transport (DAT)

DAT describes a networking infrastructure that provides the following capabilities:

- Remote Direct Memory Access (RDMA)

  RDMA directly transfers data between application memory and the memory of a remote machine without file system switches or operating system calls.
- Direct access to the network interface

  Application programs can initiate data transfer operations directly to a network interface card (NIC) without operating system intervention.

DAT improves data transfer by bypassing the operating system and moving data directly between application buffers and NIC hardware memory. As a result, DAT reduces the overall number of copies, operating system calls, and file system switches required to move data. DAT also reduces networking overhead

because the operating system does not perform data fragmentation or reassembly. The DAT-compliant NIC manages data delivery to and from the network, so protocol processing is performed at the hardware level.

Understanding the DAFS service implementation

DAFS client-server implementation

The DAFS service is implemented as a client-server solution. The DAFS server runs on a NetApp filer and is integrated into the Data ONTAP 6.4 software. The client is any system that runs a DAFS client. The DAFS client uses the DAFS protocol over a DAT connection to access files maintained by a NetApp filer running the DAFS server. This release supports the VI/IP networking infrastructure for the client-server DAT connection.

Multiprotocol file access

File systems can be mounted simultaneously using DAFS, CIFS, and NFS. Files can be shared concurrently between DAFS, CIFS, and NFS.

Transport requirements

DAFS is not a transport protocol for SnapMirror, SNMP, or NDMP; TCP/IP is still required.

Configuring the DAFS service

About the dafs setup command

The `dafs setup` command configures DAFS settings and starts the DAFS service. You can reconfigure DAFS settings at any time by running the `dafs setup` command.

Information required by the dafs setup command

The `dafs setup` command prompts you to configure the following settings:

- Network devices that use DAFS—The names of the VI/IP network interfaces that you want to configure to work with DAFS
- Filer volumes or qtrees to export—Filer volumes and qtrees that are exported via the DAFS service
- Cluster network devices that use DAFS—The names of the cluster partner VI/IP network interfaces that you want to configure to work with DAFS
- Authentication for privileged clients—This sets the initial value of `options dafs.auth_none_enable`. For details about this option, see "Configuring DAFS with options" on page 289.
- Idle timeout for sessions—The length of time in seconds that a session can be idle before a session ends, or times out

Before you use the dafs setup command

If the DAFS service is already running and you want to change existing settings with the `dafs setup` command, the following commands are helpful in displaying the current DAFS service settings and states:

Command	Information displayed
`ifconfig -a`	A list of network devices, including the VI network interface names. These are the network device names that you recorded when you gathered the required DAFS configuration information.
`dafs status -n`	The state of the DAFS service and a list of all VI NICs that use the DAFS service and their current states.

Command	Information displayed
`dafs export list`	A list of all DAFS exports with detailed information, including the exported filer paths.
`options dafs`	A list of current DAFS service settings, including the idle timeout value and whether the AUTH_NONE authentication is enabled.

Using the dafs setup command

To configure DAFS on your filer, complete the following steps.

Step	Action	
1	Enter the following command to check if the DAFS service is running. **`dafs status -n`** **Result:** The filer console displays the status of the DAFS service and a list of VI NICs and their current states.	
2	**If...**	**Then...**
	The DAFS service is running	On the filer console, enter the following command: **`dafs stop`** **Result:** The DAFS service stops. Proceed to Step 3.
	The DAFS service is not running	Proceed to Step 3.
3	On the filer command line, enter the following command: **`dafs setup`** **Note** Whenever you run `dafs setup`, all previous VI NIC and DAFS export settings are erased and you must reset them to configure your filer for the DAFS service. If you want to exit `dafs setup` and configure your filer later, press Ctrl-C.	

Step	Action

4	Respond to the `dafs setup` command prompts using the information for your configuration.
	
	Example: The following lines show example `dafs setup` command prompts:
	`filer> dafs setup`
	`Configure DAFS access to the filer.`
	
	`Configure network devices to use with DAFS.`
	`  Enter a network device for DAFS, or 'q' when done: [q]` **`e4`**
	`  Adding e4...`
	`  Enter a network device for DAFS, or 'q' when done: [q]` **`q`**
	`Configured DAFS attached devices.`
	
	`Configure cluster partner network devices to use with DAFS.`
	`  Partner network devices are used during cluster failover to maintain`
	`  DAFS service. They may be the same as the primary DAFS device(s)`
	`  or a dedicated subset of available devices.`
	
	`  Enter a partner network device for DAFS, or 'q' when done: [q]` **`q`**
	`Configured DAFS attached partner devices.`
	
	`Configure DAFS exports.`
	`  Do you want to export all filer volumes through DAFS: [n]` **`n`**
	`  Enter a filer volume/path to export, or 'q' when done: [q]` **`/vol/vol1`**
	`  Enter a filer volume/path to export, or 'q' when done: [q]` **`q`**
	`Do you want privileged clients to require authentication: [y]` **`y`**
	`Enter idle timeout for sessions, or 0 to disable timeout: [7200]` **`7200`**
	`Enabling DAFS services...`
	`Fri Nov  1 14:46:39 GMT [filer: dafs.server.startOk:info]: DAFS service started`
	`Fri Nov  1 14:46:39 GMT [filer: dafs.server.attachNic:info]: DAFS service attaching e4`
	`You are changing option dafs.auth_none.enable which applies to both members of the cluster in takeover mode.`
	`This value must be the same in both cluster members prior to any takeover or giveback, or that next takeover/giveback may not work correctly Fri Nov  1 14:46:39 GMT [filer: reg.options.cf.change:warning]: Option dafs.au`
	`filer>`
	
	Result: The `dafs setup` command enables and starts the DAFS service and attaches VI NICs to the service. The value of `options dafs.enable` is set to `on`. No reboot is required.

Step	Action

5	After you complete `dafs setup` and the DAFS service is enabled, verify that DAFS service is running and all known VI NICs are attached by entering the following command on the filer console:
	`dafs status -n`
	Result: The display shows the state of the DAFS service and a list of all VI NICs that use the DAFS service and their current states.
	Example:
	`filer> dafs status -n`
	`DAFS service is started`
	`NIC: e4 Device: e4 State: ATTACHED`
	`NIC: p_e4 Device: e4 State: INACTIVE PARTNER`
	`NIC: e5 Device: e5 State: ATTACHED`
	In these examples, the `dafs setup` command created the partner network interface p_e4 for cluster failover. The partner devices are attached, but remain in the inactive partner state until a takeover occurs.

Starting and stopping the DAFS service

When to stop and start the DAFS service

The `dafs setup` command starts the DAFS service. During regular operations, you may temporarily want to stop the DAFS service for

- Filer maintenance, for example, rearranging filer disks
- VI NIC maintenance
- Using the `dafs setup` command to reconfigure DAFS properties

For filer maintenance, it is important to stop the DAFS service (with the `dafs stop` command) and disable any other currently running protocols, for example, CIFS or NFS. After performing the maintenance, you can run `dafs start` to enable the DAFS protocol and enable any other desired protocols. When you start the DAFS service, DAFS VI NICs are automatically reattached.

You can also enable the DAFS service to run persistently (for example, to always start at boot time) or disable the DAFS service from running. To perform these operations, you use the `options dafs.enable` command.

About starting and stopping the DAFS service

To temporarily start and stop the DAFS service during filer operation, you use the `dafs start` and `dafs stop` commands. The command syntax and functions are as follows:

- The `dafs start` command starts the DAFS service if it is not already running and attaches the DAFS service to any VI NICs that are configured for the DAFS service.
- The `dafs stop` command stops the DAFS service if it is running. Prior to stopping the DAFS service, this command stops any DAFS sessions on DAFS-attached VI NICs. You must stop the DAFS service (if it is running) before running the `dafs setup` command.

For more information, see the na_dafs (1) man page.

Starting the DAFS service

To start the DAFS service, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`dafs start`
	Result: The DAFS service is started and any VI NICs that are configured to work with DAFS are attached.
	Example:
	`filer> dafs start`
	`Thu Aug 30 15:26:57 GMT [dafs.server.startOk:info]: DAFS service started`
	`Thu Aug 30 15:26:57 GMT [dafs.server.attachNic:info]: DAFS service attaching e4`
	`Thu Aug 30 15:26:58 GMT [dafs.server.attachNic:info]: DAFS service attaching e5`
	In this example, the DAFS service was started and the VI NICs e4 and e5 were attached to the DAFS service.

Stopping the DAFS service

To stop the DAFS service, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`dafs stop`
	Result: The DAFS service is stopped, including any DAFS sessions on DAFS-attached VI NICs.
	Example:
	`filer> dafs stop`
	`Thu Aug 30 15:27:21 GMT [dafs.server.shutdown:info]: DAFS service shutdown`
	In this example, the DAFS service was stopped, but will start if the the `dafs start` command is issued, or if the filer is rebooted.

About persistently enabling and disabling the DAFS service

To persistently enable or disable the DAFS service, you can use the `options dafs.enable` command. The command syntax and functions are as follows:

- The `options dafs.enable on` command starts the DAFS service if it is not already running and additionally attaches the DAFS service to any VI NICs that are configured for the DAFS service. The DAFS service is enabled to run persistently. It automatically starts each time the filer is rebooted.
- The `options dafs.enable off` command stops the DAFS service if it is running. Prior to stopping the DAFS service, this command stops any DAFS sessions on DAFS-attached VI NICs. The DAFS service is disabled persistently. It does not start after rebooting the filer system.

For more information, see the na_options(1) man page.

Persistently enabling the DAFS service

To persistently enable the DAFS service, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`options dafs.enable on`
	Result: The DAFS service is started and any VI NICs that are configured to work with DAFS are attached.
	Example:
	`filer> options dafs.enable on`
	`Thu Aug 30 15:28:53 GMT [dafs.server.startOk:info]: DAFS service started`
	`Thu Aug 30 15:28:53 GMT [dafs.server.attachNic:info]: DAFS service attaching e4`
	`Thu Aug 30 15:28:54 GMT [dafs.server.attachNic:info]: DAFS service attaching e5`
	`filer>`
	In this example, the DAFS service was started and the VI NICs e4 and e5 were attached to the DAFS service.

Persistently disabling the DAFS service

To persistently disable the DAFS service, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`options dafs.enable off`
	Result: The DAFS service is stopped, including any DAFS sessions on DAFS-attached VI NICs.
	Example:
	`filer> options dafs.enable off`
	`Thu Aug 30 15:28:50 GMT [dafs.server.shutdown:info]:`
	`DAFS service shutdown`
	In this example, the DAFS service was stopped, and will not restart if the filer is rebooted. The service can be restarted with the `options dafs.enable on` command or the `dafs start` command.

Attaching and detaching network devices

About attaching and detaching network devices

When the `dafs setup` command starts the DAFS service, it attaches the DAFS service to configured VI NICs. You can also add newly configured VI NICs or remove VI NICs while the DAFS service is running.

To attach and detach VI NICs for the DAFS services, you use the `dafs attach` and `dafs detach` commands. The command syntax and functions are as follows:

- The `dafs attach` *`vinic`* command attaches the DAFS service to the specified VI NIC. This command can also designate a failover VI NIC as a partner interface for a clustered pair of filers. For more information on clusters, see the *Storage Management Guide*.
- The `dafs detach` *`vinic`* command detaches the DAFS service from the specified *vinic*.

Attaching a device needs to be done only once. The DAFS service remembers which NICs are attached and automatically attaches itself to the same set of NICs on the next system reboot or `dafs start` command. An attached NIC can be in an inactive state. This typically applies to CFO partner devices that are attached but remain inactive until a CFO takeover occurs.

You can also add newly configured VI NICs or remove VI NICs while the DAFS service is running with the `dafs attach` or `dafs detach` commands.

For more information, see the man page na_dafs(1).

Attaching the DAFS service to VI NICs

To attach the DAFS service to a specific VI NIC, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	dafs attach *vinic* [-d *physical_device*] [-l *listen_address*][-p]
	vinic specifies the logical interface name of the VI NIC to be attached to the DAFS service.
	`-d` *physical_device* specifies the physical device name of the VI NIC. This name depends upon the specific hardware installed in your filer. The logical *vinic* interface name and the physical device name can be the same. If the physical device name is not specified, then by default the physical device name is the same as the logical *vinic* interface name.
	`-l` *listen_address* is the VI net address and discriminator where the DAFS server listens for requests. This is an advanced option and NetApp suggests that you use the defaults for this option. See "Default options for the dafs attach command" on page 259.
	`-p` indicates that the VI NIC is attached as a partner interface for cluster failover (CFO).
	Result: The DAFS service is attached to the specified VI NIC.
	Example:
	`filer> dafs attach e4`
	`filer>`
	In this example, the local interface e4 is attached to the DAFS service. The logical name e4 is the same as the physical device name and it is unnecessary to specify the `-d` option.
	Example for failover VI NIC as a partner interface:
	`filer> dafs attach pe4 -d e4 -p`
	`filer>`
	In this example, the CFO partner interface pe4 is attached to the DAFS service. The logical name of the CFO partner VI NIC is pe4, and the physical device is e4. The `-p` option causes pe4 to be attached as a CFO partner interface.

Default options for the dafs attach command

The following defaults are applied if you do not specify options for the `dafs attach` command:

- If the device (`-d`) option is not specified, then by default the physical device name is the same as the logical *vinic* interface name
- If the listen address (`-l`) option is not specified, the following defaults are used:
  - For the local interface, the default consists of the VI net address of the VI NIC, followed by a colon ":" and the discriminator called "DAFS." An example is "`-l 10.20.29.184:DAFS`."
  - For the CFO partner interface indicated with the `-p` option, the default consists of the VI net address of the VI NIC, followed by a colon ":" and the discriminator called "pDAF." An example is "`-l 10.20.29.184:pDAF`."

Detaching VI NICs from the DAFS service

To detach the DAFS service from a specific VI NIC, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`dafs detach vinic`
	vinic specifies the logical name of the VI NIC to be detached from the DAFS service.
	Example: The DAFS service is detatched from the specified VI NIC:
	`filer> dafs detach e4`
	`filer> dafs detach pe4`
	`filer>`
	In this example, the local interface e4 and the partner interface pe4 are detached from the DAFS service.

Understanding session authentication

About session authentication

Session authentication involves authenticating a client user's access for either a trusted or untrusted session. A DAFS client requests one of the following types of user access to authenticate with the filer:

- AUTH_NONE—This access bypasses authentication and can be used by a trusted user only. This is equivalent to root access. By default, AUTH_NONE is disabled.
- AUTH_TEXT—This access uses a traditional user name and password authentication.
- AUTH_GSS—This access uses GSS Host Key or Kerberos authentication.
- AUTH_DEFAULT—This access uses the default filer credentials and is similar to anonymous FTP.

Viewing filer credentials

To view the default set of filer credentials, complete the following steps.

Step	Action

1	On the filer command line, enter the following command:
	`options dafs.default_uid`
	Result: The value of the DAFS default UID (user ID) is displayed.
	Example:
	`filer> options dafs.default_uid`
	`dafs.default_uid  122`

2	On the filer command line, enter the following command:
	`options dafs.default_gid`
	Result: The value of the DAFS default GID (group ID) is displayed.
	Example:
	`filer> options dafs.default_gid`
	`dafs.default_gid  122`

Types of authentication sessions

There are two types of authenticated sessions:

- A trusted session where the client user's access has more than one set of credentials. AUTH_NONE and AUTH_TEXT access are for a trusted session.
- An untrusted session where the user's access has only one set of credentials. AUTH_DEFAULT, AUTH_TEXT, and AUTH_GSS access are for an untrusted session.

To determine whether AUTH_TEXT is for a trusted or untrusted session, the DAFS service does the following:

- If the user name and password are in the filer's user name database, then the session is trusted. The `useradmin` command administers the filer's user name database.
- If the user name and password are in NIS or the etc/passwd file, then the session is untrusted.
- If the user name and password are not in the filer's user name database, NIS, or etc/passwd file, then the user fails authentication and is not allowed access to the filer.

Using the dafs setup command to configure authentication

When you configure the DAFS service with the `dafs setup` command, you can enable or disable the filer's AUTH_NONE access. If AUTH_NONE is enabled, then all clients communicating with that filer have unlimited access to the filer. Only privileged clients that are trusted to have multiple sets of credentials should have AUTH_NONE access.

Note

AUTH_NONE access should be used in an environment where networks are secure and clients and filers are physically secured.

Enabling authentication for DAFS Database Accelerator clients

The DAFS Database Accelerator requires AUTH_NONE enabled in order to access files on the filer. To enable AUTH_NONE authentication, complete the following step.

Step	Action
1	On the filer console, enter the following command: **`options dafs.auth_none.enable on`**

About the dafs kerberos command

Data ONTAP 6.4 can participate in a Kerberos v5 realm. MIT Kerberos and Windows 2000 domains are supported for those clients that use Kerberos authentication. AUTH_GSS Host Key, AUTH_GSS Kerberos, AUTH_TEXT, and AUTH_DEFAULT can be used regardless of the setting for `dafs.auth_none.enable`. Setting up Kerberos does not preclude HOSTKEY, TEXT, or DEFAULT authentication. The `dafs kerberos` command is used to enable, disable, and configure the Kerberos environment.

Example:

```
filer> dafs kerberos
Usage:
dafs kerberos [ setup | enable | disable ]
- Perform DAFS kerberos setup.
```

Enabling and configuring Kerberos authentication for DAFS clients

To enable Kerberos authentication, use the `dafs kerberos setup` command as shown in the following steps.

Step	Action	
1	Enter the following command to check if the DAFS service is running: **`dafs status -n`** **Result:** The filer console displays the status of the DAFS service.	
2	**If....**	**Then...**
	The DAFS service is running	On the filer console, enter the following command: **`dafs stop`** **Result:** The DAFS service stops. Proceed to Step 3.
	The DAFS service is not running	Proceed to Step 3.

Step	Action

3	On the filer command line, enter the following command:
	`dafs kerberos setup`
	Respond to the prompts using information for your environment.
	Example: The following is an example of `dafs kerberos setup` for a Windows KDC authenticated environment:
	`filer> dafs kerberos setup`
	`Enter the kerberos realm:` **`realm.netapp.com`**
	`Enter the password for filer principal (1 - 63 characters):`
	`Supported encryption algorithms are:`
	`(1) ENCTYPE_DES_CBC_CRC`
	`(2) ENCTYPE_DES_CBC_MD5`
	`Enter the encryption algorithm: [2]` **`1`**
	`GSS/Kerberos authentication for DAFS has been setup successfully`
	`Please make sure the principal " dafs/filer.realm.netapp.com@realm.netapp.com " has been added to the`
	`KDC`
	Note
	The MD5 encryption algorithm is the default for MIT Kerberos. CRC is the default for Windows 2000 KDCs. For specific information, see the documentation for your Kerberos KDC.
	Result: The `dafs kerberos setup` command enables Kerberos authentication for those clients that support it and starts the DAFS protocol with Kerberos authentication enabled.

Enabling and disabling Kerberos authentication

You can turn Kerberos authentication on or off at any time by issuing the `dafs kerberos enable` or `dafs kerberos disable` command at the filer command prompt.

Examples:

```
filer> dafs kerberos disable
GSS/Kerberos authentication for DAFS has been disabled

filer> dafs kerberos enable
GSS/Kerberos authentication for DAFS has been enabled
```

Using Host Key authentication

The filer can authenticate DAFS clients via GSS using a 128-byte key value that is known to both the client and the filer. This key is generated by the client and stored in the file /etc/dafs_key on the root volume of the filer. The format of this file is as follows:

```
[client name] [filer name] [128B key]
```

Note

For instructions on generating the `[128B key]` value, see your DAFS client documentation.

Managing exports for DAFS service

About DAFS exports

A DAFS export is a uniquely named object associated with a filer path (export point). DAFS clients access filer volumes or qtrees through DAFS exports. The DAFS exports are visible to DAFS clients, allowing them either read-only or read-write access. Volumes and qtrees must be exported before they can be accessed.

Each DAFS export has an associated access control list (ACL) based on the DAFS client host. For more information about host access, see "Managing client host access to DAFS exports" on page 270.

About managing DAFS exports

To manage DAFS exports for the DAFS service, you use the `dafs export` command. The command syntax and functions are as follows:

- The `dafs export create [-r] export_name path` command creates a new DAFS export with the specified *export_name* and an associated filer *path.* The `-r` argument creates a read-only export. The default access is read-write.
- The `dafs export destroy export_name` command destroys the DAFS export with the specified *export_name.*
- The `dafs export list [export_name]` command lists the all the DAFS exports and their properties or one DAFS export specified with *export_name*.
- The `dafs export modify -r|-w export_name` command modifies the read-only and read-write attributes.
- The `dafs export enable export_name` command makes the DAFS export specified with *export_name* available after a previous `dafs export disable` command.
- The `dafs export disable export_name` command disables the DAFS export specified with *export_name.* A disabled export has configuration information preserved on the filer but is no longer available for DAFS clients to access. The `dafs export enable` command makes the named DAFS export available to DAFS clients.
- The `dafs export all` command reexports all named DAFS exports for the current host.

For more information, see the na_dafs_export(1) man page.

About naming DAFS exports

The `dafs export create` command enables you to specify a name for each exported path. When you create DAFS exports with this command, use unique prefixes for each export name.

If you create exports with matching prefixes, then you cannot delete a specific export. The `dafs export destroy` command deletes all exports with matching prefixes in their names. For example, if you create three exports named exp1, exp12, and exp13, then all the export names have an "exp1" prefix. If you use the `dafs export destroy` command to delete exp1, the command deletes all three exports.

Creating a DAFS export

To create and name a new DAFS export, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`dafs export create [-r] export_name path`
	export_name is the DAFS export name.
	path is the associated filer path.
	`-r` creates a read-only export.
	The default access is read-write.
	Result: A new DAFS export is created.
	Examples:
	`filer> dafs export create data1 /vol/vol3`
	`filer>`
	In this example, the data1 export with the /vol/vol3 filer volume path is created.
	`filer> dafs export create data2 /vol/vol3/qtree2`
	`filer>`
	In this example, the data2 export with the /vol/vol3/qtree2 filer qtree path is created.

Destroying a DAFS export

To destroy a named DAFS export, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	dafs export destroy *export_name*
	export_name is the DAFS export name.
	Result: The named DAFS export is destroyed. Access by DAFS clients is removed when the next client access is attempted.
	Example:
	`filer> dafs export destroy data1`
	`filer>`
	In this example, the data1 export is destroyed.

Listing DAFS exports

To list the DAFS exports and their properties, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	dafs export list [*export_name*]
	export_name specifies a single DAFS export name; otherwise all DAFS exports are displayed.
	Result: The display shows a list of DAFS exports.
	Example:
	`filer> dafs export list`
	`Name: data1`
	`  Export point: /vol/data1`
	`  Enabled: true`
	`  Read/Write Mode: read_write`
	`Name: vol0`
	`  Export point: /vol/vol0`
	`  Enabled: true`
	`  Read/Write Mode: read_write`
	`2 export(s) found`

Modifying a DAFS export

To modify a DAFS export's read-only or read-write attributes, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`dafs export modify -r|-w export_name`
	export_name is the DAFS export name.
	`-r` marks the DAFS export as a read-only export.
	`-w` marks the DAFS export as a read-write export.
	Result: The named DAFS export's attributes are modified.
	Example:
	`filer> dafs export modify -r data1`
	`filer>`
	In this example, the data1 export is modified to be a read-only export.

Enabling a disabled DAFS export

To enable a previously disabled DAFS export, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`dafs export enable export_name`
	export_name is the DAFS export name.
	Result: The named DAFS export is enabled and is now available to be accessed by DAFS clients.
	Example:
	`filer> dafs export enable data1`
	`filer>`
	In this example, the data1 export is reenabled.

Disabling a DAFS export

To disable a DAFS export, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`dafs export disable export_name`
	export_name is the DAFS export name.
	Result: The named DAFS export is disabled and can no longer be accessed by DAFS clients. The configuration information for the disabled export is preserved on the filer. This is useful when you perform maintenance on your filer.
	Example:
	`filer> dafs export disable data1`
	`filer>`
	In this example, the data1 export is disabled.

Reexporting all DAFS exports

To reexport all the DAFS exports, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`dafs export all`
	Result: All DAFS exports are reexported for the current host.
	Example:
	`filer> dafs export all`
	`filer>`

Managing client host access to DAFS exports

About ACLs for DAFS exports

Each DAFS export has an associated access control list (ACL) that controls how the client host can access it. The following are the entry types in an ACL:

- ACCESS—This gives default read and write access to the DAFS export for all or specified hosts.
- DENY—This denies access to the DAFS export for specific hosts.

  If `-n` is specified in the `dafs access` command, then the ACL entry for the host is DENY.

The default is an empty export ACL indicating that all client hosts can access a DAFS export. This default provides the best performance. The host-based ACLs on the DAFS exports give or deny access at the export level. ACLs are subject to the DAFS exports restrictions, for example, a read-only access mode or a disabled DAFS export.

For more information about DAFS exports, see "Managing exports for DAFS service" on page 265.

Note

When you update ACLs, the changes take effect immediately.

About managing host access to DAFS exports

To manage client host access to DAFS exports, use the `dafs access` command. The command syntax and functions provided are as follows:

- The `dafs access add` command adds a host to the ACL for either all DAFS export ACLs or a specified ACL.
- The `dafs access delete` command deletes a host from the ACL for either all DAFS export ACLs or a specified ACL.
- The `dafs access set` command replaces the contents of the ACL with the list of hosts provided on the command line for either all DAFS exports or a specified DAFS export.
- The `dafs access get` command displays the ACL for either all DAFS exports or a specified DAFS export.

Note

The ACL entries ACCESS and DENY are mutually exclusive. An export cannot have allowed and disallowed host(s) in the same ACL.

Adding a client host to a DAFS export ACL

To add a client host to a DAFS export ACL, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	dafs access add [-n] -h *hostname* [-e *export_name*] -p *path*]
	`-n` indicates that the ACL entry for the host is DENY.
	hostname is the client's host name. You must use the host names of VI NICs on the client. A sample host name is db_server.
	export_name is the DAFS export name. If *export_name* is not specified, then the host is added to all DAFS exports.
	Result: The host name is added to either all DAFS export ACLs or a specified ACL.
	Examples:
	`filer> dafs access add -h client -e data1`
	`filer>`
	`filer> dafs access add -n -h client_ext`
	`filer>`
	In the first example, client is added to the ACL for the data1 export. In the second example, a DENY entry for client_ext is added to all DAFS exports to deny this client access to all DAFS exports.

Deleting a client host from a DAFS export ACL

To delete a host from a DAFS export ACL, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`dafs access delete [-n] -h hostname [-e export_name|-p path]`
	`-n` indicates that the ACL entry for the host is DENY.
	hostname is the client's host name. You must use the host name of a VI NIC on the client. A sample host name is client.
	export_name is the DAFS export name. If *export_name* is not specified, then the host is deleted from all DAFS exports.
	Result: The host name is deleted from either all DAFS export ACLs or a specified ACL.
	Examples:
	`filer> dafs access delete -h client`
	`filer>`
	`filer> dafs access delete -h client_1 -e data1`
	`filer>`
	In the first example, client is deleted from all DAFS exports. In the second example, client_1 is deleted from the data1 DAFS export.

Replacing the ACL for a DAFS export

To replace the ACL for a DAFS export with a list of host names, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`dafs access set [-n] -h host_list [-e export_name|-p path]`
	`-n` indicates that the ACL entry for the hosts is DENY.
	host_list is a list of client VI host names, delimited by colons (:). A sample host name is dbclient.
	export_name is the DAFS export name. If *export_name* is not specified, then *host_list* replaces all ACLs for all DAFS exports.
	Result: *host_list* replaces the ACL for either all DAFS exports or a specified DAFS export. If *host_list* is not specified, then the ACL is made empty.
	Example:
	`filer> dafs access set -h dbclient:dbclient -e data1`
	`filer>`
	In this example, the host list dbclient:dbclient replaces the ACL for data1 export.
	Note
	If you want to reset the host list for an export name to the default (empty) list, enter the following command:
	`dafs access set -h "" -e export_name`

Displaying the ACL for a DAFS export

To display the ACL for a DAFS export, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`dafs access get [-e export_name | -p path]`
	export_name is the DAFS export name. If *export_name* is not specified, then the ACLs for all DAFS exports are displayed.
	Result: The display shows the ACLs for either all DAFS exports or a specified DAFS export. The displayed ACL shows the DAFS export name, the ACL entry type, and the client host name.
	Example:
	`filer> dafs access get`
	`qtree1  deny client1`
	`qtree2  <none>`
	`vol1    access  client2`
	◆ The DAFS export qtree1 denies access to client1; all other DAFS clients are allowed access.
	◆ The DAFS export qtree2 allows all DAFS clients access; none are denied access. This is the default ACL.
	◆ The DAFS export vol1 allows access for client2; all other DAFS clients are denied access.

Measuring DAFS performance

About measuring DAFS performance

To measure the performance of the DAFS service, you use the `sysstat -d` and `dafs stats` commands. The command syntax and functions provided are as follows:

- The `sysstat -d` command reports aggregated filer performance statistics, including current CPU utilization; I/O usage for network, disk, and tape; and DAFS statistics with the DAFS default output format.
- The `dafs stats` command gathers statistics on individual DAFS server operations for a specified time interval. During the time interval, the following statistical information is reported for each operation:
  - The number of count operations that occurred
  - The percentage of total operations represented
  - The rate (operations per second) of the operation
  - The time per operation in milliseconds

For more information, see the man pages for na_sysstat(1) and na_dafs(1).

Displaying filer statistics

To display filer statistics that include the DAFS statistics with the DAFS default output format, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`sysstat -d`
	Result: The display shows the filer statistics, including the DAFS operations (in place of the HTTP operations).
	Example:
	`sysstat -d:`
	`CPU NFS CIFS DAFS  Net kB/s   Disk kB/s   Tape kB/s Cache`
	`                    in   out   read write read write  age`
	`8%  0   0    2395 316  9930   8     0    0     0     51`
	`8%  0   0    2385 315  9894   8     0    0     0     51`
	Note
	When invoked with no arguments, `sysstat` prints a new line of statistics every 15 seconds. Press Ctrl-C or set the interval count (`-c`) to stop the `sysstat` reporting. The `sysstat -x` option displays the extended format output.

Gathering DAFS statistics only

To gather DAFS server operation statistics over a specified time interval, complete the following steps.

Step	Action

1	At the filer command line, enter the following command:
	`dafs stats start`
	Result: The filer begins gathering statistics about the DAFS service. This command enables statistics gathering only and no statistics are displayed.
	Note
	The time interval is determined by the elapsed time between entering the `dafs stats start` command and a `dafs stats stop` command. If you start gathering statistics and then enter the `dafs stats list -z` command, the statistics counters are reset to zero. In this case, the time interval is the elapsed time between resetting the counters to zero and entering the `dafs stats stop` command.
	Example:
	`filer> dafs stats start`
	`filer>`

Step	Action

2	To view DAFS statistics as they are gathered, enter the following command at the filer command line:
	dafs stats list
	Result: The display shows the statistics for the elapsed time between entering the `dafs stats start` command and the `dafs stats list` command.
	Note
	This command displays an intermediate result. Statistics continue to be gathered until the `dafs stats stop` command is run. You can zero the statistics counters with the `-z` option. For more information, see the na_dafs(1) man page.
	Example:
	`filer> dafs stats list`
	`DAFS Statistics, gathered for 23 seconds, ops 38439`
	`Operation Count Percent Ops/sec Msec/op`
	`--------- ------- ------- ------- ------`
	`Client Connect-Authenticate 1 0 0 1`
	`Register Credentials 1 0 0 2`
	`Get root handle 1 0 0 4`
	`Lookup 2 0 0 1`
	`Open 1 0 0 1`
	`Read Inline 38431 99 1670 2`
	`Remove 1 0 0 2`
	`Write Inline 1 0 0 1`
	In this example, a list of DAFS statistics is displayed. A total of 23 seconds have elapsed since the gathering of DAFS statistics began.

Step	Action

3	To stop the statistics gathering, enter the following command at the filer command line:
	`dafs stats stop`
	Result: The display shows the statistical results for the elapsed time between entering the `dafs stats start` command and the `dafs stats stop` command. If you reset the statistics counters to zero with the `dafs stats list -z` command, then the results are for the elapsed time between resetting the counters and entering the `dafs stats stop` command.
	Example:
	`filer> dafs stats stop`
	`DAFS Statistics, gathered for 31 seconds, ops 56483`
	`Operation Count Percent Ops/sec Msec/op`
	`--------- ------ ------- ------- ----`
	`Client Connect-Authenticate 1 0 0 1`
	`Register Credentials 1 0 0 2`
	`Get root handle 1 0 0 4`
	`Lookup 2 0 0 1`
	`Open 1 0 0 1`
	`Read Inline 56475 99 1821 2`
	`Remove 1 0 0 2`
	`Write Inline 1 0 0 1`
	In this example, the statistics gathering has ended and the final results are displayed. A total of 31 seconds elapsed for gathering statistics.

Explanation of DAFS statistics

The `dafs stats` command gathers statistics on requests and operations, grouping them by the request. The statistics show the following:

- Name of the request (operation)
- Count of each operation
- Percent of total operations
- Number of operations per second over the elapsed time interval
- Average time in milliseconds to execute an operation

Displaying DAFS status

About displaying DAFS status

To display the status of the DAFS service, NICs, and sessions, you use the `dafs status` command. The command syntax and functions provided are as follows:

- The `dafs status -v` command displays in detail the state of the DAFS service.
- The `dafs status -vn` command displays the details about the VI NICs associated with the DAFS service.
- The `dafs status -n` command displays a list of all VI NICs associated with the DAFS service and their current states.
- The `dafs status -sn` command displays the state of all sessions associated with VI NICs that use the DAFS service.
- The `dafs status -sr` command displays the state of the DAFS service, basic information for client sessions, and current DAFS requests.
- The `dafs status -sv` command displays the state of the DAFS service and details of the client sessions.
- The `dafs status -a` command displays the state of the DAFS service, VI NIC status, and client session status.

For more information, see the na_dafs(1) man page.

Displaying the state of DAFS service

To display in detail the state of the DAFS service, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`dafs status -v`
	Result: The display shows the details (verbose option) of the state of the DAFS service for server sessions, server requests, and data transfers.
	Example:
	`filer> dafs status -v`
	`DAFS service is started`
	`Sessions: 2, disconnected: 0, max disconnected: 32`
	`Server Requests: curr: 0, limit: 4096, pend: 0, max pend: 128`
	`Max requests per session: 1024, NICs active: 1`
	`Bytes in: 2668, bytes out: 10964`
	`Direct bytes in: 0, direct bytes out: 0`

Explanation of dafs status -v output

The `dafs status -v` command displays the following output:

- State of the DAFS service
- Current number of sessions
- Number of disconnected sessions
- Maximum number of disconnected sessions

The server request information shows the following:

- Current number of requests being processed
- Current limit on dispatching new requests
- Number of pending requests due to limit restriction
- Maximum limit of pending requests
- Maximum number of requests that a session can have at one time

Note

The dispatch limit on starting new requests prevents DAFS from overloading the filer. Requests are pended if they cannot be dispatched.

- Number of attached VI NICs

The data transfer information shows the following:

- Count of bytes in and out of the filer.
- Count of direct bytes in and out of the filer. Direct bytes are RDMA transfers.

Displaying detailed status of VI NICs

To display detailed status for all VI NICs associated with the DAFS service, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`dafs status -vn`
	Result: The display shows the details about all the VI NICs associated with the DAFS service. This command is useful for checking VI NIC IP addresses and verifying whether the NICs are attached to the DAFS service.
	Note
	If you have multiple VI NICs and want to view the same level of detail status for one specific NIC, enter the following command:
	`dafs status -vn` ***`vinic`***
	vinic is a VI NIC network interface name, for example, `dafs status -vn e5`
	Example:
	`filer> dafs status -vn`
	`DAFS service is started`
	`NIC: e4 Device: e4 State: ATTACHED`
	`Listen Address: 10.40.132.109:DAFS`
	`Requests in: 27`
	`Bytes in: 2668, bytes out: 10964`
	`Direct bytes in: 0, direct bytes out: 0`
	`Request daemons: 2`
	`Descriptor Cache: T:2, S:2, M:0, L:0, RM:0`
	`Cache Hit/Miss: T:24/2, S:5/8, M:2/2, L:0/0 RM:0/0`
	`Send pending highwater: 1, limit: 921`
	`Request flowcontrol: 0, flowcontrol timeout: 0`

Explanation of dafs status -vn output

The `dafs status -vn` command displays the following output:

- NIC transport information—Specifies the listen address.
- Data transfer information—Requests in, bytes in and out, direct bytes in and out, and request daemons.
- Cache and flow control information—Each NIC has a cache of VI descriptors reused between sessions. There are five cache buckets based on the following sizes: T (tiny), S (small), M (medium), L (large), and RM (descriptor packets received in multiple segments). The statistics show the number of descriptors in the cache and the hits and misses for each bucket.

Displaying list of VI NICs

To display the list of all VI NICs associated with the DAFS service, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	`dafs status -n`
	Result: The display shows the state of the DAFS service and a list of all VI NICs that use the DAFS service and their current states.
	Example:
	`filer> dafs status -n`
	`DAFS service is started`
	`NIC: e4  Device: e4  State: ATTACHED`
	`NIC: e5  Device: e5  State: ATTACHED`

Displaying sessions associated with VI NICs

To display all sessions associated with VI NICs that use the DAFS service, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	dafs status -sn
	Result: The display shows the state of the DAFS service and the status of all sessions associated with VI NICs that use the DAFS service.
	Example:
	`filer> dafs status -sn`
	`DAFS service is started`
	`NIC: e5 Device: e5 State: ATTACHED`
	`Sessions`
	`--------`
	`Session ID State Flags`
	`300100008b700002 CONNECTED auth,trust,endian-diff`
	`500100008b700002 CONNECTED auth,trust,endian-diff`
	`2 session(s) found`
	
	`NIC: e9 Device: e9 State: ATTACHED`
	`Sessions`
	`--------`
	`Session ID State Flags`
	`400100008b700002 CONNECTED auth,trust,endian-diff`
	`1 session(s) found`
	Note
	If you have multiple VI NICs and want to view the session status for one specific NIC, enter the following command:
	dafs status -sn vinic
	vinic is a VI NIC interface name, for example, `dafs status -sn e5`

Displaying all sessions and DAFS requests

To display all sessions associated with the DAFS service and any DAFS requests for a session, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	dafs status -sr
	Result: The display shows the state of the DAFS service, basic information for client sessions, and current DAFS requests.
	Example:
	`filer> dafs status -sr`
	`DAFS service is started`
	`Sessions`
	`--------`
	`Session ID State Flags`
	`010000008b700002 CONNECTED auth,trust,endian-diff`
	`Request: Read Inline, state: RESPONSE_PENDING,`
	`stream/seq: 0/220`
	`1 session(s) found`
	Note
	If you have multiple sessions and want to view the requests active in a specific session, enter the following command:
	dafs status -sr *session*
	session is a session ID, for example,
	`dafs status -sr 010000008b700002`

Displaying client sessions

To display the details of all client sessions associated with the DAFS service, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	dafs status -sv
	Result: The display shows the state of the DAFS service and details of all client sessions associated with the DAFS service.
	Example:
	`filer> dafs status -sv`
	`DAFS service is started`
	`Sessions`
	`--------`
	`0400000048720002 CONNECTED auth,trust,endian-diff,rescache`
	`Remote name: dbclient-viip0`
	`Client identifier: DTS-CLIENT2`
	`Idle: 12 second(s)`
	`Max request size: 14000, max response size: 5192`
	`Max requests: 100, current requests: 0, nreq: 1`
	`Registered Credentials, max: 10, current: 1.`
	`0300000048720002 CONNECTED auth,trust,endian-diff,rescache`
	`Remote name: dbclient-viip0`
	`Client identifier: DTS-CLIENT`
	`Idle: 109 second(s)`
	`Max request size: 14000, max response size: 5192`
	`Max requests: 100, current requests: 0, nreq: 1`
	`Registered Credentials, max: 10, current: 1.`
	`2 session(s) found`

Explanation of dafs status -sv output

The `dafs status -sv` command displays the following output:

- State of the DAFS service
- Session ID—A cluster-wide unique ID
- State of the session—CONNECTED or DISCONNECTED

  You might also see the following interim states:

  - IDLE—The session was created but the connection was not completed.

- DISCONNECT PENDING—The filer is clearing any remaining operations before disconnecting and removing the session.
- MORIBUND—The session is being disconnected as a result of an error. The filer is clearing remaining operations before disconnecting the session. After operations are cleared, the session state is DISCONNECTED.

- Remote host name (as resolved from transport)
- Client identifier—Supplied by the client in DAFS_PROC_CLIENT_CONNECT calls
- Idle time
- Negotiated limits for requests
- Number of sessions found

The session flags (or state flags) list the properties for a particular session. The possible properties are

- back—The session has a back channel.
- rdmard—The session has an RDMA_READ channel.
- cksum—Checksums are enabled for the session.
- auth—The session is authenticated.
- trust—The session is trusted.
- endian-diff—The session is endian different, indicating byte swapping to maintain byte order.
- rescache—The session has the response cache enabled. The filer holds completed client operations in the response cache. If a client session terminates, the client can check the response cache upon reconnecting for a record of completed operations.

Displaying all DAFS status information

To display all DAFS status information for the DAFS service, VI NICs, and client sessions, complete the following step.

Step	Action

1	At the filer command line, enter the following command:
	dafs status -a
	Example: The following lines shows the output of `dafs status -a`:
	`DAFS service is started.`
	`  Sessions: 0, disconnected: 0, max disconnected: 32`
	`  Server Requests: curr: 0, limit: 1024, pend: 0, max pend: 512`
	`  Max requests per session: 50, NICs active: 1`
	`  Bytes in: 0, bytes out: 0`
	`  Direct bytes in: 0, direct bytes out: 0`
	`NIC: e4  Device: e4  State: ATTACHED`
	`  Listen Address: 192.168.1.2:DAFS`
	`  Requests in: 0`
	`  Bytes in: 0, bytes out: 0`
	`  Direct bytes in: 0, direct bytes out: 0`
	`  Request daemons: 2`
	`  Descriptor Cache: T:0, S:1, M:0, L:0, RM:0`
	`  Cache Hit/Miss: T:0/0, S:0/1, M:0/0, L:0/0 RM:0/0`
	`  Send pending highwater: 0, limit: 3686, exceed: 0`
	`  Request flowcontrol: 0, flowcontrol timeout: 0`
	`Sessions`
	`--------`
	`Session ID  State Flags`
	`0 session(s) found`
	`Channels`
	`--------`
	`Channel  Type  State  Session  NIC  Flags`
	`0x361e1b60 PRIMARY  UNUSED  <none>  e4  0x189`
	`  VI: 0x361b18b8, Remote name:`
	`  flags: has_vi,endian_diff,dyn_nreq,recv_multi`
	`  max_requests:0 nreq:0 sendsize:0 recvsize:0`
	`  post_send:0, post_recv:0 loaned_recv:0`
	`1 channel(s) found`
	`Memory Stats`
	`Memory pool: allocs: 471 frees: 441 current: 30`
	`VI memory: allocs: 2 frees: 1 current: 1`
	`Channels : allocs: 2 frees: 1 current: 1 total mem: 276`
	`Requests : allocs: 0 frees: 0 current: 0 unused: 0 total mem: 0`
	`Sessions : allocs: 0 frees: 0 current: 0 unused: 0 total mem: 0`
	`MBUF pool: allocs: 0 used: 0 unused: 0`
	`WAFL buffers: steal: 0 unsteal: 0 current: 0`
	`VI Register: 0, Deregister: 2`
	`VI Non-WAFL Register: 0, Deregister: 0`
	`Max large descriptors cached/NIC: 100`

Configuring DAFS with options

Use options to configure DAFS

To configure DAFS, use options.

The DAFS options are shown in the following table.

Option	Description
`dafs.auth_none.enable`	Enables use of the DAFS AUTH_NONE (no security) authentication. This is disabled (off) by default.
`dafs.checksums.enable`	Enables use of DAFS-level checksums for DAFS requests and responses. You can override this option on a per-NIC basis using the `dafs` command.
`dafs.default_gid`	Sets the default numeric group ID (`gid`) to be used when DAFS client credentials are supplied as CRED_DEFAULT or AUTH_DEFAULT.
`dafs.default_uid`	Sets the default numeric user ID (`uid`) to be used when DAFS client credentials are supplied as CRED_DEFAULT or AUTH_DEFAULT.
`dafs.enable`	Determines whether the DAFS server starts by default on a filer.

Option	Description
dafs.idle_timeout	Sets the time in seconds that a DAFS session can be idle (no requests made) before it is a candidate for disconnection by the filer. A value of 0 disables the idle timeout mechanism. The default idle timeout is 7200 seconds (2 hours). The maximum idle timeout is 65535 seconds (about 18 hours).
dafs.max_disconnected_sessions	Sets the maximum number of abnormally disconnected client sessions that the DAFS server will save until recovery can be made. If the number of abnormally disconnected client sessions exceeds this limit, the oldest disconnected session is removed.
dafs.max_pending_requests_server	The total number of active concurrent requests.
dafs.max_request_size	Sets the maximum size of a request buffer that a single session can request during session negotiation. Normally, you should not change this value from the default, but it can be set to allow (or disallow) clients with exceptionally large DAFS requests to execute. The value, expressed in bytes, limits the total request size, including DAFS operation and generic request headers.
dafs.max_requests	Sets the maximum number of concurrent DAFS requests that a single session can request during session negotiation.

Option	Description
dafs.max_requests_server	The total number of requests that will be queued waiting for a process slot.
dafs.max_response_size	Sets the maximum size of a response buffer that a single session can request during session negotiation. This should not be changed from the default, but you can set it to allow (or disallow) clients with exceptionally large DAFS responses to execute. The value, expressed in bytes, limits the total response size, including DAFS operation and generic response headers.
dafs.num_requestd	Sets the default number of DAFS request-processing servers to start for each NIC. You can override this option on a per-NIC basis using the dafs command.
dafs.percent_memory	The percentage of main memory DAFS is allowed to cache.
dafs.session_reuse_threshold	Sets the minimum number of seconds a DAFS session must be idle before it becomes eligible for disconnection to reclaim resources needed for other active sessions. The value is expressed in seconds. The default is 600. A value of 0 indicates that no session should ever be disconnected.

File Sharing Between NFS and CIFS 7

About this chapter

This chapter describes how Data ONTAP enables NFS and CIFS clients to share files and how you can optimize Data ONTAP to share files quickly and without errors.

Topics in this chapter

This chapter discusses the following topics:

- "Understanding NFS and CIFS file naming" on page 294
- "Understanding file locking between protocols" on page 296
- "Understanding read-only bits" on page 297
- "Controlling CIFS access to symbolic links" on page 299
- "Optimizing NFS directory access for CIFS clients" on page 312
- "Preventing CIFS clients from creating uppercase file names" on page 314
- "Accessing CIFS files from NFS clients" on page 315
- "Installing SecureShare Access" on page 324
- "Changing UNIX permissions and DOS attributes from Windows" on page 325

Understanding NFS and CIFS file naming

About file naming conventions in multiprotocol environments

File naming conventions depend on both the network clients' operating systems and the file-sharing protocols. The operating system and the file-sharing protocols determine the

- Length of a file name
- Characters a file name can use
- Case-sensitivity of a file name

Length of file names

Data ONTAP supports the following file name lengths:

- Maximum of 255 characters for NFS clients and CIFS clients that support the PC long file name format
- Maximum of 8-character file names and 3-character file name extensions for CIFS clients that support the 8.3 format, such as MS-DOS and Windows 3.*x* clients

Characters a file name can use

If you are sharing a file between clients on different operating systems, you should use characters that are legal to both operating systems. For example, if you use UNIX to create a file, don't use a colon (:) in the file name because the colon is an illegal MS-DOS character. Because restrictions on legal characters vary from one operating system to another, refer to the documentation for your client operating system for more information about prohibited characters.

Case-sensitivity of a file name

File names are case-insensitive, but case-preserving, for CIFS clients and case-sensitive for NFS clients.

For example, if a CIFS client creates Spec.txt, both CIFS and NFS clients display the file name as Spec.txt. If a CIFS user later tries to create spec.txt, the user's application instructs the user to choose another name because, to the CIFS client, that name currently exists. If an NFS user later creates a file named spec.txt, NFS and CIFS clients display the file name differently, as follows:

- On NFS clients, you see both file names as they were created, Spec.txt and spec.txt, because file names are case-sensitive.

- On CIFS clients, even those that support the PC long file name format, you see Spec.txt and Spec~1.txt. Data ONTAP creates the Spec~1.txt file name to differentiate the two files.

Creating lowercase file names

You can set an option to have all new file names created in lowercase text for better compatibility between 16-bit applications and some UNIX tools. This option is turned on by default, which does not force lowercase file names.

To set lowercase text for all new file names, complete the following step.

Step	Action
1	Enter the following command: `options cifs.save_case off`

How Data ONTAP creates file names

Data ONTAP creates and maintains two file names for files in any directory that has access from a CIFS client: the original long name and a file name in 8.3 format. Data ONTAP generates an 8.3 format file name as follows:

1. It truncates the original file name to six characters, if the file name exceeds six characters.
2. It appends a tilde (~) and a number or letter to the name, to ensure that the file name is unique. 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.

Example: If an NFS client creates a file named specifications.html, the 8.3 format file name created by Data ONTAP is specif~1.htm. If this name already exists, Data ONTAP uses a different number at the end of the file name. For example, if the NFS client creates another file named specifications_new.html, the 8.3 format of specifications_new.html is specif~2.htm.

Understanding file locking between protocols

About file locking

File locking is a method used by client applications to prevent a user from accessing a file previously opened by another user. How Data ONTAP locks files depends on the protocol of the client.

- If the client is an NFS client, locks are advisory.
- If the client is a CIFS client, locks are mandatory.

Which NFS operations fail

Because of the difference between the locks, some attempts by an NFS client to access a file opened by a CIFS application fail.

The following occurs when an NFS client attempts to access a file locked by a CIFS application:

- File manipulation operations, such as `rm`, `rmdir`, and `mv`, by an NFS client on a file opened by a CIFS client application can cause the NFS application to fail. This only occurs in mixed or NTFS qtrees.
- NFS read and write operations will be denied by CIFS deny-read and deny-write open modes, respectively.
- NFS write operations will also fail when the written range of the file is locked with an exclusive CIFS bytelock.

Exception: One exception to the enforcement of locks set by CIFS clients on the filer is when the filer runs the `dump` command. The `dump` command ignores the file lock set by a CIFS client that prevents read access to parts of a file or the entire file. Ignoring the file lock enables the filer to back up all files.

Note

If an attempt by an NFS client to access a file opened by a CIFS application fails, you can use the `cifs terminate` command to disconnect the session that has the open file that you want to access. You can determine which session has that file open using the `cifs sessions *` command or Server Manager. If you terminate a CIFS session, the client may receive errors.

Understanding read-only bits

What a read-only bit is

The read-only bit is a bit that is set on a file-by-file basis to reflect whether a file is writable or read-only.

Which clients set a read-only bit

CIFS clients that use MS-DOS and Windows can set a per-file read-only bit. NFS clients do not set a per-file read-only bit because NFS clients do not have any protocol operations that use a per-file read-only bit.

When Data ONTAP can set a read-only bit

Data ONTAP sets a read-only bit on a file when a CIFS client that uses MS-DOS or Windows creates that file. Data ONTAP can also set a read-only bit when a file is shared between NFS clients and CIFS clients because some software, when used by NFS clients and CIFS clients, requires that the read-only bit reflects whether the file is writable.

How Data ONTAP keeps appropriate permissions on shared files

For Data ONTAP to keep the appropriate read and write permissions on a file shared between NFS clients and CIFS clients, it treats the read-only bit according to the following rules:

- 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.
- 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 the read-only bit for a file is turned on and an NFS client attempts to discover permission bits for the file, the 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 the read-only bit for a file 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 root.

Note

Changes to a file's permissions appear immediately on CIFS clients, but might not appear immediately on NFS clients if the NFS client enables attribute caching.

Deleting files with the read-only bit set

Windows does not allow you to delete a file with the read-only bit set. Some multiprotocol source control applications require UNIX delete semantics; files for these applications with the read-only bit set also cannot be deleted.

To allow deletion of files using UNIX delete semantics when the read-only bit is set, complete the following step.

Step	Action
1	Enter the following command: `options cifs.perm_check_ro_del_ok on` This option is turned off by default.

Controlling CIFS access to symbolic links

What a symbolic link is

A symbolic link is a special file created by NFS clients that points to another file or directory. A symbolic link is, in some respects, similar to a "shortcut" in the Windows environment.

Kinds of symbolic links

There are two kinds of symbolic links:

- 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.

CIFS clients cannot create symbolic links, but they can follow the symbolic links created by NFS clients.

For detailed information

The following sections discuss the ways you can control how CIFS clients follow symbolic links:

- "Understanding how CIFS clients interact with symbolic links" on page 300
- "About Map entries" on page 303
- "About Widelink entries" on page 304
- "About disabling share boundary checking for symbolic links" on page 306
- "Redirecting absolute symbolic links" on page 308
- "Preventing access by CIFS clients to cyclic directory structures" on page 311

Understanding how CIFS clients interact with symbolic links

Enabling CIFS clients to follow symbolic links

To allow CIFS clients to follow symbolic links on the filer, you must turn on the `cifs.symlinks.enable` option. This option is enabled by default on the filer. If this option has been disabled, you may need to reenable it.

To enable CIFS access to symbolic links, complete the following step:

Step	Action
1	Enter the following command to enable CIFS access to symbolic links: **`options cifs.symlinks.enable on`**

When `options cifs.symlinks.enable` is turned on, CIFS clients will directly follow relative symbolic links to destinations in the same share.

You must complete additional steps if you want to allow CIFS clients to follow these symbolic links:

- Absolute symbolic links

  Since the destination of an absolute symbolic link depends on the way the UNIX mount is set up, CIFS clients need additional information to interpret absolute symbolic links.
- Relative symbolic links to destinations on the same filer that are outside the share in which the relative symbolic link is located

  By default, the filer will not allow a CIFS client to follow a symbolic link outside the share to which the CIFS client is authenticated.

The following section describes the options for setting up the way CIFS clients behave when they encounter relative and absolute symbolic links on the filer.

Ways CIFS clients can interact with symbolic links

You can specify how you want CIFS clients to interact with symbolic links by using one or more of the following options:

- Create Map entries in the /etc/symlinks.translations file (absolute symbolic links only)
- Create Widelink entries in the /etc/symlinks.translations file (absolute symbolic links only)

- Disable NT share boundary checking for symbolic links (relative and absolute symbolic links)

Use the following table to help determine which options you want to implement. The table shows for each option the types of destinations that symbolic links will be able to point to.

Symbolic link destination can be...	Options for CIFS access to symbolic links		
	Map entries	Widelink entries	No share boundary check
The same share on the same filer	X	X	X
Another share on the same filer		X	X
A non-shared area of the same filer			X
A share on another filer		X	
A share on another CIFS server or a desktop PC		X	

For more information about each of these options, see the following sections:

- "About Map entries" on page 303
- "About Widelink entries" on page 304
- "About disabling share boundary checking for symbolic links" on page 306

Why you should avoid symbolic links to files

You should prevent CIFS clients from following symbolic links that point to files because Data ONTAP can update the wrong files. The wrong files may be updated because many CIFS client applications 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.

When client applications perform these operations, if the original file was targeted directly by a symbolic link, that file would be stored in the directory where the symbolic link was, and the renamed symbolic link would point to the original file rather than to the updated file.

Note

CIFS clients following symbolic links to directories, rather than to individual files, do not experience this problem.

About Map entries

What Map entries are

Map entries are used to redirect absolute symbolic links on the filer. You create Map entries in the /etc/symlink.translations file. Map entries allow CIFS clients to follow absolute symbolic links to target destinations within the same share.

Requirements for Map entries

Map entries have the following requirements:

- In order resolve an absolute symbolic link, there must be a Map entry in the /etc/symlink.translations file that determines the destination of the link.
- The destination of the absolute symbolic link must be in the same share on the same filer as the absolute symbolic link.

Using Map entries

When you use Map entries to redirect absolute symbolic links, NT share security is preserved for both the symbolic link and the destination because they are in the same share. If you have both Map entries and Widelink entries in the symlink.translations file, the filer will use the first matching entry it finds.

For more information

For information about how to create Map entries to redirect absolute symbolic links, see “Redirecting absolute symbolic links” on page 308.

About Widelink entries

What Widelink entries are

Widelink entries are another way to redirect absolute symbolic links on the filer. You create Widelink entries in the /etc/symlink.translations file. Widelink entries allow CIFS clients to follow absolute symbolic links to target destinations either on the same filer or outside the filer. This is enabled on a per-share basis.

Requirements for Widelink entries

Widelink entries have the following requirements:

- The share in which the absolute symbolic links are located must be enabled for wide symbolic links.
- In order to resolve an absolute symbolic link, there must be a Widelink entry in the /etc/symlink.translations file that determines the destination of the link.
- The destination of the Widelink entry must be one of the following:
  - The same share as the symbolic link
  - Another share on the same filer
  - A share on another filer
  - A share on another CIFS server or desktop PC
- The CIFS client must have client-side support for Microsoft Distributed File System (DFS). Windows NT and later clients support DFS by default.

Using Widelink entries

To follow Widelink entries, the CIFS client automatically requests and receives a DFS referral from the filer to establish an authenticated connection with the target share. This preserves NT share security for both the symbolic link and the destination. Once the connection is established, the CIFS client can make new requests directly to the target share or server, thereby increasing performance.

If you have both Map entries and Widelink entries in the symlink.translations file, the filer will use the first matching entry it finds.

Limitations

Widelink entries have the following limitations:

- If the destination of the wide symbolic link is a file, it will be displayed as a directory in directory listings. The system API for opening the file will

correctly follow the wide symbolic link, but this may confuse certain applications. To avoid this problem, create a wide symbolic link that resolves to a directory, rather than a file.

- Windows 95, Windows 98, and Windows ME clients cannot follow a wide symbolic link to another wide symbolic link.
- Windows NT clients cannot display or modify ACLs in a share enabled for wide symbolic links. This restriction does not apply to Windows 2000 and later clients.
- Wide symbolic links cannot direct a client to a non-shared area on the destination machine.

For more information

For information about how to create Widelink entries to redirect symbolic links, see "Redirecting absolute symbolic links" on page 308. For information about how to enable a share for wide symbolic links, see "Sharing directories on the filer" on page 95.

About disabling share boundary checking for symbolic links

How disabling share boundary checking affects symbolic links

When you disable share boundary checking for symbolic links, CIFS clients can follow symbolic links anywhere on the same filer. This behavior is set on a per-share basis and affects both relative and absolute symbolic links.

Requirements

Disabling share boundary checking for symbolic links has the following requirements:

- The share in which the symbolic links are located must be set to `nosymlink_strict_security`.
- In order to resolve an absolute symbolic link, there must be a Map entry in the /etc/symlink.translations file that determines the destination of the link.
- The destinations for relative symbolic links and for mapped absolute symbolic links may be in any shared or non-shared area of the filer.

Limitations

Disabling share boundary checking for symbolic links has the following limitations:

- Relative symbolic links cannot be used to span volumes; you must use absolute symbolic links.
- Symbolic links cannot be followed off the filer to other systems.
- NT share security is:
  - Preserved for the symbolic link itself because the CIFS client has to authenticate to connect to the share in which the symbolic link is located.
  - Preserved for the destination of the symbolic link only if the destination is in the same share.
  - Not preserved for the destination of the symbolic link if the destination is outside the share because the CIFS client does not have to authenticate to the destination (which may or may not be a CIFS share).

Note

If you disable share boundary checking for symbolic links, be sure to secure any areas of the filer that you do not want users to access. This is necessary because there is nothing to restrict a user from creating a symbolic link to any path on the filer.

For more information

For more information about disabling share boundary checking for symbolic links, see “Sharing directories on the filer” on page 95.

Redirecting absolute symbolic links

About redirecting absolute symbolic links

NFS clients interpret the file system location represented by an absolute symbolic link based on how the file systems are mounted on the client. CIFS clients do not have access to NFS clients' mount information.

To allow CIFS clients to follow absolute symbolic links on the filer, you must redirect the absolute symbolic link so that CIFS clients can interpret the file system location represented by the absolute symbolic link. You can redirect absolute symbolic links by creating entries in the /etc/symlink.translations file. The /etc/symlink.translations file performs the same role on the filer as automounter tables on UNIX servers.

Ways to redirect symbolic links

You can redirect absolute symbolic links on the filer using one or both of the following methods:

- Create Map entries in the /etc/symlink.translations file
- Create Widelink entries in the /etc/symlink.translations file

For details about Map entries, see "About Map entries" on page 303. For information about Widelink entries, see "About Widelink entries" on page 304.

Creating Map entries

To create Map entries to redirect absolute symbolic links in a CIFS environment, complete the following steps.

Step	Action
1	Open the /etc/symlink.translations file for editing.

Step	Action

2	Enter one or more lines in the file using the following format:
	`Map template result`
	template is used to match absolute symbolic links.
	result is a filer path that is substituted for the matching absolute symbolic link.
	Examples:
	`Map /u/users/charlie/* /home/charlie/*`
	`Map /temp1/* /vol/vol2/util/t/*`

Creating Widelink entries

To create Widelink entries, complete the following steps.

Step	Action
1	Open the /etc/symlink.translations file for editing.

2	Enter one or more lines in the file using the following format:
	`Widelink template [@qtree] result`
	template is the UNIX path name.
	result is the CIFS UNC path name.
	@qtree can be optionally specified to allow multiple entries in different qtrees to have the same template value.
	Examples: In the following examples, *result* uses CIFS path name syntax, with backslashes as separators, and allows an embedded space. The wildcard character (*) in the template path name represents zero or more characters, including the slash character (/). In the result path name, the wildcard character represents text from the corresponding match in the template path name.
	`Widelink /eng/proj/* @/vol/vol2 \\filer\hw\proj\*`
	`Widelink /eng/proj/* \\filer\sw\proj\*`

How the filer uses Map and Widelink entries

To allow CIFS clients to follow absolute symbolic links, the filer searches the entries in the etc/symlink.translations file in sequential order until a matching entry is found or the lookup fails. The filer uses the first matching entry it finds to generate a path to the destination. Therefore, it is important to put the most restrictive entries first to prevent premature mapping errors.

Example: This example shows how to list Map entries. /u/home/* is more specific than /u/*.

```
Map /u/home/* /vol/vol2/home/*
Map /u/* /vol/vol0/*
```

Example: This example shows how to list Widelink entries.

```
Widelink /u/docs/* \\filer\engr\tech pubs\*
Widelink /u/* \\filer\engr\*
```

Preventing access by CIFS clients to cyclic directory structures

About cyclic directory structures

A cyclic directory structure is one that includes a symbolic link that uses a “dot” or “dot-dot” component to make a reference to a directory at the same level or higher in the same tree.

Because CIFS clients don’t understand symbolic links, a CIFS client following a cyclic directory structure goes deeper and deeper into the tree until Data ONTAP reaches its maximum path length. When this happens, Data ONTAP returns an error.

Disabling CIFS access

To disable CIFS access to cyclic directory structures, complete the following step.

Step	Action
1	Enter the following console command: `options cifs.symlinks.cycleguard on` This option is turned on by default.

Optimizing NFS directory access for CIFS clients

About NFS directory accessibility

When you first install Data ONTAP, directories created by NFS clients are in pre-Data ONTAP 4.0 format and directories created by CIFS clients are in Unicode format. CIFS directories are immediately accessible to NFS clients, but NFS directories are not immediately accessible to CIFS clients. To access an NFS directory, CIFS clients must convert the NFS directory format from pre-Data ONTAP 4.0 format to Unicode format. This is done automatically by the filer when the access occurs.

Optimizing CIFS client access to an NFS directory

You can optimize CIFS client access to an NFS directory by performing the following tasks to eliminate the time lag created by converting an NFS directory to Unicode format:

- Change Data ONTAP to create only Unicode-formatted directories, thereby eliminating the need to convert formats.

  Note

  You should cause Data ONTAP to create directories in Unicode format immediately after installing Data ONTAP to ensure that all directories on the filer are in Unicode format.

- Configure Data ONTAP to convert existing NFS directories to Unicode format when either CIFS clients or NFS clients access directories.

Creating Unicode-formatted directories

To cause Data ONTAP to create all directories in Unicode format, complete the following step.

Step	Action
1	Enter the following command: `options wafl.create_ucode on`

Converting to Unicode format

By default, Data ONTAP converts a directory format from pre-Data ONTAP 4.0 to Unicode only when a CIFS client accesses that directory. You may want to shorten the time it takes to convert every directory to Unicode format by

- Using an NFS client to distribute files in directories that contain more than 50,000 files among a greater number of subdirectories
- Converting a directory format when either a CIFS client or an NFS client accesses that directory

To distribute files to subdirectories and to convert directories to Unicode format when they are accessed from either CIFS or NFS, complete the following steps.

Step	Action	
1	**If...**	**Then...**
	You have directories that contain more than 50,000 files	**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.
	You have directories that contain fewer than 50,000 files	Go to Step 2.
2	Enter the following command: **`options wafl.convert_ucode on`** on causes Data ONTAP to convert directories accessed by CIFS or NFS to Unicode format.	

Preventing CIFS clients from creating uppercase file names

How CIFS clients rename NFS files

Older, 16-bit CIFS clients that open and save files change the file name by changing the lowercase or mixed-case characters to all uppercase characters. You can prevent these uppercase file names by forcing Data ONTAP to store CIFS file names using lowercase characters.

Forcing lowercase file names

To prevent CIFS clients from creating uppercase file names, complete the following step.

Step	Action
1	Enter the following command: `options cifs.save_case off`

Accessing CIFS files from NFS clients

How NFS clients access CIFS files

Data ONTAP uses Windows NT File System (NTFS) security semantics to determine whether a UNIX user, on an NFS client, has access to a file in a mixed or NTFS qtree. Data ONTAP does this by converting the user's UNIX User ID (UID) into a CIFS credential, then using the CIFS credential to verify that the user has access rights to the file. A CIFS credential consists of a primary Security Identifier (SID), usually the user's Windows NT user name, and one or more group SIDs that correspond to Windows NT groups of which the user is a member.

The time Data ONTAP takes converting the UNIX UID into a CIFS credential can be from tens of milliseconds to hundreds of milliseconds because the process involves contacting a domain controller. Data ONTAP maps the UID to the CIFS credential and enters the mapping in a WAFL credential cache to reduce the verification time caused by the conversion. You can control the WAFL credential cache to further reduce the time Data ONTAP takes to verify rights. You can also monitor WAFL credential cache statistics to help you determine what CIFS credentials are currently in the WAFL credential cache.

For detailed information

The following sections discuss tasks you can perform to manage the WAFL credential cache:

- "Adding mapping entries to the WAFL credential cache" on page 316
- "Deleting mapping entries from the WAFL credential cache" on page 317
- "Setting how long mapping entries are valid" on page 318
- "Monitoring WAFL credential cache statistics" on page 319
- "Managing mapping inconsistencies" on page 321

Adding mapping entries to the WAFL credential cache

About adding entries

You can add mapping entries to the WAFL credential cache at any time. Normally, this is not necessary because entries are created automatically as the filer is accessed.

The best way to add entries is in a script that loads the WAFL credential cache with entries at boot time. This immediately puts the entries in the WAFL credential cache rather than waiting for Data ONTAP to create the entries in the course of accessing the files.

Caution

If you add more entries than the 10,000 allowed, the older entries are deleted.

Prerequisites

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

Adding an entry

To add an entry to the WAFL credential cache, complete the following step.

Step	Action
1	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 mapping entries from the WAFL credential cache

About deleting entries

You can delete entries from the WAFL credential cache at any time. You delete entries to avoid periods of time when security changes are not seen. Such periods of time can occur if you change a user's rights, such as removing a user from a group, and a mapping for that user already exists in the WAFL credential cache. In this case, the user has access to files that should no longer be available until the entry in the WAFL credential cache times out automatically. The default for timing out automatically is 20 minutes.

Prerequisites

You must have the name, and optionally the IP address, of the entry you want to delete from the WAFL credential cache.

Deleting an entry

To delete an entry from the WAFL credential cache, complete the following step.

Step	Action
1	Enter the following command: **`wcc -x name`** *name* is one of the following specifications: ◆ `-s` followed by the Windows NT user name or group name found in the CIFS credential **Caution** If *name* is the name of a group, this procedure deletes all members of that group from the WAFL credential cache. ◆ `-u` followed by the UNIX name found in the CIFS credential 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. If you do not specify *name*, all entries are deleted. **Example:** `wcc -x -u jdoe -i 10.100.4.41`

Setting how long mapping entries are valid

Setting appropriate length of validity

Using the default time or increasing the time that the CIFS credential remains in the WAFL credential cache after Data ONTAP updates it improves performance. Performance is improved because Data ONTAP doesn't have to take the time to create a CIFS credential to verify access to a file.

The disadvantage of increasing the time that CIFS credentials remain in the WAFL credential cache is that if you change a user's access rights, the change does not take effect until Data ONTAP updates the WAFL credential cache. In this case, the user might have rights to a file to which you have just denied access.

If you do not expect problems of this type, you can increase the time that the credential entry is valid. If you need to see access right updates as they occur and slower performance is not an issue, you can use a smaller value than the default.

Setting the length of validity

To set how long each WAFL credential cache entry is valid, complete the following step.

Step	Action
1	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 20,160. The default value is 20.

Monitoring WAFL credential cache statistics

About monitoring WAFL credential cache statistics

By monitoring WAFL credential cache statistics, you can view

- What entries are currently cached
- The UNIX UID-to-CIFS credential mapping

This information is useful when you need to know what entries are in the WAFL credential cache or what the access rights are for users listed in the entries.

Displaying WAFL credential cache statistics

To display statistics about the WAFL credential cache, complete the following step.

Step	Action
1	Enter the following command: **`wcc -d uname`** *uname* is the UNIX name of a user. Omit *uname* to list all credential entries in the WAFL credential cache. You can get more detailed information by appending `-v` to the command line. You can have up to three instances of the `-v` option (`-vvv`) per command.

Sample output

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

```
wcc -d
tday (UID 10350) from 10.121.4.41 => NT-DOMAIN\tday*

Total WCC entries: 3; oldest is 127 sec.
Total Administrator-privileged entries: 1
 * indicates members of "BUILTIN\Administrators" group
```

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

```
wcc -dvv
```

```
jdoe (UID 1321) from 10.121.4.41 => NT-DOMAIN\jdoe
        ***************
        UNIX uid = 1321

        NT membership
                NT-DOMAIN\jdoe
                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
        ***************
tday (UID 10350) from 10.121.4.41 => NT-DOMAIN\tday*
        ***************
        UNIX uid = 10350

        NT membership
                NT-DOMAIN\tday
                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
        ***************
bday (UID 1219) from 10.121.4.41 => NT-DOMAIN\bday
        ***************
        UNIX uid = 1219

        NT membership
                NT-DOMAIN\bday
                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
```

Managing mapping inconsistencies

Problems found in mapping inconsistencies

If a user cannot access a file that should be accessible, the problem can be one of the following:

- You granted access recently and the WAFL credential cache does not have the new mapping entry.

  You can determine mapping inconsistencies between recently granted rights and the WAFL credential cache by comparing CIFS credential mappings. You can display mapping results for either the user's UNIX name or Windows NT account name.
- The NFS client could not obtain CIFS credentials.

  You can determine whether an NFS client can perform a CIFS login to the filer by tracing CIFS logins.
- Depending on the NFS client, it may be necessary to wait for the NFS attribute cache to time out before changes to the CIFS credential take effect.

Determining mapping inconsistencies with a UNIX name

To determine mapping inconsistencies with a UNIX name, complete the following steps.

Step	Action
1	Display the current CIFS credential mapping of a UNIX name by entering 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. You can have up to three instances of the `-v` option (`-vvv`) per command.
2	Note the CIFS credential information.

Step	Action
3	Display the CIFS credential mappings in the WAFL credential cache by entering the following command: **`wcc -d`** You can get more detailed information by appending `-v` to the command line. You can have up to three instances of the `-v` option (`-vvv`) per command.
4	Compare the two CIFS credential mappings.
5	If entries in the WAFL credential cache are incorrect, delete them by entering the following command: **`wcc -x`** ***`name`*** See "Deleting mapping entries from the WAFL credential cache" on page 317 for details about deleting entries.

Determining mapping inconsistencies with a Windows NT account name

To determine mapping inconsistencies with a Windows NT account name, complete the following steps.

Step	Action
1	Display the current CIFS credential mapping of a Windows NT account name by entering the following command: **`wcc -s uname`** *uname* is the Windows NT account name: ◆ 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. You can have up to three instances of the `-v` option (`-vvv`) per command.
2	Note the CIFS credential information.

Step	Action
3	To display information about all connected users, enter the following command: `cifs sessions -s` Locate the user's information in the output.
4	Compare the two CIFS credential mappings.
5	If the CIFS credential mappings are different, disconnect the client by entering the following command: `cifs terminate workstation` When the client reconnects, the CIFS credential mappings will be correct.

Tracing CIFS logins

To monitor an attempt by an NFS client to obtain a CIFS credential, complete the following step.

Caution

Use CIFS login tracing carefully because every CIFS login is traced and constant use can result in multiple console and log messages.

Step	Action
1	Enter the following command: `options cifs.trace_login on | off` Use on to enable or off to disable CIFS login tracing.

Installing SecureShare Access

About SecureShare Access

For multiprotocol environments, Data ONTAP provides SecureShare Access. SecureShare Access makes it easy for Windows users to change UNIX file permissions on the filer.

SecureShare Access appears as a tab in the Properties dialog box of a set of files you select if at least one of the files is a UNIX file. To use SecureShare Access, follow the instructions in "Changing UNIX permissions and DOS attributes from Windows" on page 325.

Installing SecureShare Access

To install SecureShare Access from the CD-ROM, complete the following steps.

Step	Action
1	Make sure that the Data ONTAP CD-ROM is in the CD-ROM drive.
2	Click the CD-ROM drive icon.
3	Click UTILS.
4	Click SSACCESS.
5	Click SSACCESS.EXE.
6	Follow the prompts in the SecureShare Access installer program.

Changing UNIX permissions and DOS attributes from Windows

About changing UNIX permissions

Files and directories on the filer can have either Windows or UNIX-style permissions. You can change Windows-style permissions from Windows by editing Windows permissions. You can use the SecureShare Access tool to

- Change UNIX permissions in a UNIX file system from the Windows desktop
- Change DOS attributes

To use the SecureShare Access tool from a client, you must install the tool on the client. For instructions about installing the SecureShare Access tool, see "Installing SecureShare Access" on page 324.

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.

Caution

SecureShare Access has no undo feature. Use it very carefully.

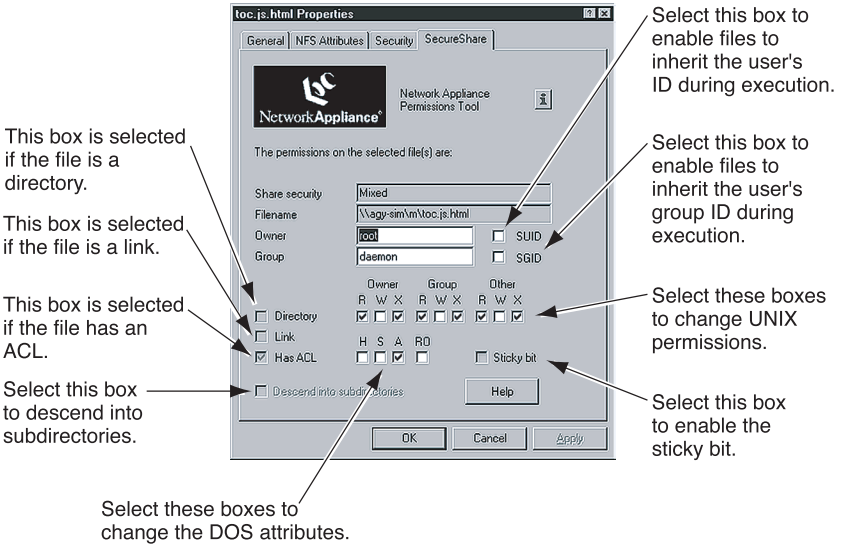
Displaying SecureShare Access

To display SecureShare Access, follow these steps.

Step	Action
1	Select the files and directories whose permissions you want to change.
2	Right-click one of the items you selected.
3	Choose Properties from the pop-up menu.
4	Click the SecureShare tab.

Changing the permissions of a single item

If you select only one item, SecureShare Access appears as follows.



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.

To change the permissions of a single item, complete the following steps.

Step	Action
1	Select an item, then display SecureShare Access by following Steps 2 through 4 in “Displaying SecureShare Access” on page 325.
2	Change permissions by selecting the appropriate boxes in the SecureShare Access window.
3	Click Apply or OK.

Changing the permissions of multiple items

In the multiple-item display, the following conditions are in effect that are different from 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.

To change the permissions of multiple items, complete the following steps.

Step	Action
1	Select several items at once, then display SecureShare Access by following Steps 2 through 4 in “Displaying SecureShare Access” on page 325.
2	Change permissions by selecting the appropriate boxes in the SecureShare Access window.
3	Click Apply or OK.

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 mail host 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.

agent

A Data ONTAP process that gathers status and diagnostic information and forwards it to NMSs.

appliance

A device that performs a well-defined function and is easy to install and operate.

ATM

Asynchronous Transfer Mode. A network technology that combines the features of cell-switching and multiplexing to offer reliable and efficient network services. ATM provides an interface between devices such as workstations and routers, and the network.

authentication

A security step performed by a domain controller for the filer's domain, or by the filer itself, using its /etc/passwd file.

autosupport A filer daemon that triggers email messages from the customer site to Network Appliance or another specified email 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.

cluster A pair of filers connected so that one filer can detect when the other is not working and, if so, can serve the failed filer data.

cluster interconnect Cables and adapters with which the two filers in a cluster are connected and over which heartbeat and WAFL log information are transmitted when both filers are running.

cluster monitor Software that administers the relationship of filers in the cluster through the `cf` command.

community A name used as a password by the SNMP manager to communicate with the filer agent.

console A terminal that is attached to a filer's serial port and is used to monitor and manage filer operation.

copy-on-write The technique for creating snapshots without consuming excess disk space.

degraded mode The operating mode of a filer when a disk is missing from the RAID array or the batteries on the NVRAM card are low.

disk ID number A number assigned by the filer to each disk when it probes the disks at boot time.

disk shelf A shelf that contains disk drives and is attached to the filer.

emulated filer A software copy of the failed filer that is hosted by the takeover filer. The emulated filer appears to users and administrators like a functional version of the failed filer. For example, it has the same name as the failed filer.

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.

failed filer The physical filer that has ceased operating. It remains the failed filer until giveback succeeds.

FDDI adapter A Fiber Distributed Data Interface (FDDI) interface card.

FDDI-fiber An FDDI adapter that supports a fiber-optic cable.

FDDI-TP An FDDI adapter that supports a twisted-pair cable.

GID Group identification number.

giveback The return of identity from the virtual filer to the failed filer, resulting in a return to normal operation; the reverse of takeover.

group A group of users defined in the filer's /etc/group file.

heartbeat A repeating signal transmitted from one filer to the other that indicates that the filer is in operation. Heartbeat information is also stored on disk.

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.

LAN Emulation (LANE) The architecture, protocols, and services that create an Emulated LAN using ATM as an underlying network topology. LANE enables ATM-connected end systems to communicate with other LAN-based systems.

local filer The filer you are logged in to.

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.

mailbox disk One of a set of disks owned by each filer that is used to store the cluster state information of the filer. If that filer stops operating, the takeover filer uses the information in the mailbox disks in constructing a virtual filer. Mailbox disks are also used as file system disks.

mail host The client host responsible for sending automatic email to Network Appliance when certain filer events occur.

maintenance mode An option when booting a filer from a system boot disk. Maintenance mode provides special commands for troubleshooting hardware and configuration.

MIB Management Information Base. An ASCII file that describes the information that the agent forwards to NMSs.

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 start the application that is appropriate for the data received.

MultiStore An optional software product that enables you to partition the storage and network resources of a single filer so that it appears as multiple filers on the network.

NDMP Network Data Management Protocol. A protocol that allows Network Appliance filers to communicate with backup applications, and provides capabilities for controlling the robotics of multiple tape backup devices.

network adapter An Ethernet, FDDI, or Asynchronous Transfer Mode (ATM) adapter.

network management station See NMS.

NMS Network Management Station. A host on a network that uses third-party network management application (SNMP manager) to process status and diagnostic information about a filer.

normal mode The state of a filer when there is no takeover in the cluster.

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 An adapter card that contains the filer's NVRAM cache.

NVRAM mirror A synchronously updated copy of the contents of the filer NVRAM (Nonvolatile Random Access Memory) contents kept on the partner filer.

panic A serious error condition causing the filer to halt. Similar to a software crash in the Windows system environment.

parity disk The 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.

partner From the point of view of the local filer, the other filer in the cluster.

partner mode The method you use to communicate through the command-line interface with the virtual filer during a takeover.

PCI Peripheral Component Interconnect. The bus architecture used in newer filer models.

pcnfsd A filer daemon that permits PCs to mount filer file systems. The corresponding PC client software is called (PC)NFS.

PDC Primary Domain Controller. The domain controller that has negotiated to be, or has been assigned as, the primary authentication server for the domain.

POST Power-on self-tests. The tests run by the filer after the power is turned on.

PVC Permanent Virtual Circuit. A link with a static route defined in advance, usually by manual setup.

qtree A special subdirectory of the root of a volume that acts as a virtual subvolume with special attributes.

RAID Redundant array of independent disks. A technique that protects against disk failure by computing parity information based on the contents of all the disks in the array. NetApp filers use 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 address The full address of a disk, consisting of the disk's SCSI adapter number and the disk's SCSI ID; for example, 9a.1.

SCSI ID The number of a disk drive on the SCSI chain (0 to 6).

serial adapter An expansion card for attaching a terminal as the console on some filer models.

serial console An ASCII or ANSI terminal attached to a filer's serial port. Used to monitor and manage filer operations.

share A directory or directory structure on the filer that has been made available to network users and can be mapped to a drive letter on a CIFS client.

SID Security identifier.

snapshot An online, 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 to back up the filer to tape while the filer is in use.

SVC Switched Virtual Circuit. A connection established through signaling. The user defines the endpoints when the call is initiated.

system board A printed circuit board that contains the filer's CPU, expansion bus slots, and system memory.

takeover The emulation of the failed filer identity by the takeover filer in a cluster; the reverse of giveback.

takeover filer A filer that remains in operation after the other filer stops working and that hosts a virtual filer that manages access to the failed filer disk shelves and network connections. The takeover filer maintains its own identity and the virtual filer maintains the failed filer identity.

takeover mode The method you use to interact with a filer while it has taken over its partner. The console prompt indicates when the filer is in takeover mode.

trap An asynchronous, unsolicited message sent by an SNMP agent to an SNMP manager indicating that an event has occurred on the filer.

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.

VCI Virtual Channel Identifier. A unique numerical tag defined by a 16-bit field in the ATM cell header that identifies a virtual channel over which the cell is to travel.

vfiler A virtual filer you create using MultiStore, which enables you to partition the storage and network resources of a single filer so that it appears as multiple filers on the network.

VGA adapter An expansion card for attaching a VGA terminal as the console.

volume A file system.

VPI Virtual Path Identifier. An eight-bit field in the ATM cell header that indicates the virtual path over which the cell should be routed.

WAFL Write Anywhere File Layout. The WAFL file system was designed for the Network Appliance filer to optimize write performance.

WINS Windows Internet Name Service.

workgroup A collection of computers running Windows NT or Windows for Workgroups that is grouped for browsing and sharing.

Index

Symbols

(PC)NFS
- clients
  - authentication support for 57
  - file permissions for 60
- creating user entries with 59
- umask
  - definition of (options pcnfsd.umask) 60
  - working with files 60

/etc/cifs_nbalias.cfg file, creating entries in 69
/etc/exports file
- contents of 29
- exporting entries differently than specified 44
- exporting entries with (exportfs) 44
- how it works 29
- limitation of 30
- rules for entries in 30
- specified resources in 24
- when changes take effect 29

/etc/ftpusers file
- about 203
- editing 211
- format of 211

/etc/group file, format of 89
/etc/home_dir file, specifying directories in 122
/etc/httpd.access file
- contents of 229
- directives in 231
- editing 231
- format of 231

/etc/httpd.group file
- about 234
- contents of 229
- editing 235
- ways to create 234

/etc/httpd.hostprefixes file 236
/etc/httpd.mimetypes file 224
/etc/httpd.passwd file
- about 234
- contents of 229
- creating and editing 234

/etc/httpd.translations file
- description of 216
- fail rule, adding 223
- map rule, adding 217
- pass rule, adding 221
- redirect rule, adding 219

/etc/log/adtlog.evt file (external audit file) 161
/etc/log/cifsaudit.alf file (internal audit file) 161
/etc/log/httpd.log file
- changing file format of 243
- information in 242

/etc/netgroup file
- description of 39
- format of entries 39

/etc/passwd file
- contents of 89
- format of 89
- used by pcnfsd 57
- when created 89

/etc/shadow file, used by pcnfsd 57
/etc/symlink.translations file (symbolic link translations) 308
/etc/symlink.translations file, listing entries in 310
/etc/usermap.cfg file
- contents of 77, 78, 79
- format of 77
- guidelines for mapping names in 85
- how Data ONTAP uses 76
- increasing security in 85
- mapping NT names and UNIX names in 72, 76

A

access restrictions
- for exported resources 26
- interactions of restriction options 27
- meanings of restriction options 26

Account Migrator, SecureShare 91
account, Windows guest 94
accounts, NT guest 94
ACLs
- adding, from Server Manager 136
- changing

from command line 139
from Server Manager 135
share-level 132
displaying from command line 140
file-level, description of 141
removal of entries
from command line 140
from Server Manager 138
administration tasks, periodic 17
aliases, NetBIOS 68
anonymous FTP access, configuring 210
authentication, setting up for FTP 207

C

CIFS
credentials, client UID/GID 72
file renaming by CIFS clients, prevention of 314
filer limitations 105
forcing lowercase file naming (options cifs.save_case) 295, 314
generic UNIX account, definition of 92
matching users to directories 115
NFS accessing CIFS files 315
CIFS commands
cifs access (changes ACL) 139
cifs access (displays ACL) 140
cifs access (removes entries from ACL) 140
cifs audit clear (clears internal event log) 171
cifs audit save (updates the event log) 165
cifs broadcast (sends messages to users) 159
cifs comment (displays and changes filer description) 67
cifs nbalias (displays list of aliases) 69
cifs nbalias load (processes entries in cifs_nbalias.cfg) 69
cifs prefdc (manages preferred domain controllers) 195
cifs resetdc (reestablishes connection with a domain controller) 197
cifs restart (restarts CIFS for a filer) 190
cifs sessions (displays information about Windows 2000 clients) 181
cifs sessions (displays share and file information for users) 182
cifs sessions (displays summary session information) 181
cifs sessions (displays user authentication method) 108
cifs sessions (displays user security information) 182
cifs setup (reconfigures CIFS) 191, 193
cifs shares (creates a share) 101
cifs shares (deletes share) 104
cifs shares (displays and changes properties) 103
cifs sidcache clear (clears all cache entries) 148
cifs sidcache clear (clears cache entries for a domain) 148
cifs sidcache clear (clears specific SID entry) 149
cifs sidcache clear (clears specific user entry) 149
cifs terminate (disables CIFS for a filer) 187
cifs terminate (disconnects selected client) 186
fpolicy disable (disables file screening 152
fpolicy enable (enables file screening) 151
fpolicy ext excl add (adds excluded extensions) 154
fpolicy ext excl remove (removes excluded extensions) 155
fpolicy ext excl reset (resets to default file screening) 155
fpolicy ext excl set (replaces list of excluded extensions) 156
fpolicy ext excl show (displays list of excluded extensions) 154
fpolicy options default required (requires screening for file access) 156
fpolicy servers show default (displays screening server status) 157
fpolicy servers stop default (disconnects file screening server) 157
fpolicy show (displays file policy information) 153
options cifs.audit.autosave.file.extension counter (specifies automatic counter extensions) 169
options cifs.audit.autosave.file.extension

timestamp (specifies automatic timestamp extensions) 170
options cifs.audit.autosave.limit (specifies maximum automatic saves) 170
options cifs.audit.autosave.onsize.enable (saves automatically by log size) 166
options cifs.audit.autosave.onsize.threshold (specifies log size threshold) 166
options cifs.audit.autosave.ontime.interval (specifies a time interval for automatic saves) 167
options cifs.audit.enable (enables and disables auditing) 163
options cifs.audit.file_access_events.enable (enables and disables file access auditing) 163
options cifs.audit.logon_events.enable (enables and disables auditing of logon and logoff events) 164
options cifs.audit.logsize (specifies maximum size of internal event log) 170
options cifs.audit.saveas (saves as different event log) 165
options cifs.home_dir (specifies home directory paths) 120
options cifs.home_dir (stops offering home shares) 131
options cifs.home_dir_namestyle (specifies naming style of home directories) 124
options cifs.idle_timeout (specifies idle session timeout) 183
options cifs.netbios_aliases (specifies NetBIOS aliases for filer) 68
options cifs.netbios_over_tcp.enable (enables or disables NetBIOS over TCP) 198
options cifs.nfs_root_ignore_acl (allows access to root volume with NTFS security) 192
options cifs.oplocks.enable (enables system-wide oplock) 179
options cifs.oplocks.opendelta (changes delay time for sending breaks) 180
options cifs.perm_check_ro_del_ok (allows deletion of files with read-only bit set) 298
options cifs.perm_check_user_gid (sets GID to grant access) 134
options cifs.save_case (forces lowercase file names) 295, 314
options cifs.shutdown_msg_level (specifies which users receive shutdown messages) 189
options cifs.sidcache.enable (enables and disables SID-to-name map cache) 146
options cifs.sidcache.lifetime (changes lifetime of mapping entries) 147
options cifs.symlinks.cycleguard (disables access to cyclic directory structures) 311
options cifs.symlinks.enable (enables CIFS clients to follow symbolic links) 300
options cifs.trace_login (traces CIFS logins by NFS clients) 323
options fpolicy.enable (turns file policies on or off) 151
options wafl.default_unix_user (creates generic account) 93
CIFS protocol 103
accessing others' home directories 128
accessing root volume with NTFS security (options cifs.nfs_root_ignore_acl) 192
ACLs
file-level 141
share-level 132
adding ACL entries with Server Manager 136
audit events, list of 173
auditing
description of 161
external log file, /etc/log/adtlog.evt 161
internal log file, /etc/log/cifsaudit.alf 161
prerequisites for 161
auditing of logon and logoff events, enabling and disabling (options cifs.audit.logon_events.enable) 164
changing filer description using Server Manager 66
changing from command line (cifs access) 139
cifsconfig.cfg file 191

clearing event audit log (cifs audit clear) 171
creating generic user account (options wafl.default_unix_user) 93
creating UNIX accounts for CIFS users 90
default setting of GID 134
disabling 184
  for the filer (cifs terminate) 187
disabling access to cyclic directory structures (options cifs.symlinks.cycleguard) 311
disconnecting clients using command line (cifs terminate) 186
disconnecting clients with Server Manager 185
displaying
  ACLs, from command line (cifs access) 140
  session information 181
  share and file information for users (cifs sessions) 182
  summary session (cifs sessions) 181
  user security information (cifs sessions) 182
displaying file policy information (fpolicy show) 153
event detail displays
  description of 173
  from UNIX 175
  from Windows 174
  lost record 176
  unsuccessful file access 175
event log
  automatic counter extensions 169
  automatic file naming 168
  automatic saves 166
  automatic timestamp extensions 169
  clearing internal log (cifs audit clear) 171
  enabling automatic saves by log size (options cifs.audit.autosave.onsize.enable) 166
  how updated 162
  saving as a different log file (options cifs.audit.saveas) 165
  size and format of 161
  specifying a time interval for automatic saves (options cifs.audit.autosave.ontime.interval) 167
  specifying counter extensions (options cifs.audit.autosave.file.extension counter) 169
  specifying log size threshold (options cifs.audit.autosave.onsize.threshold) 166
  specifying maximum automatic saves (options cifs.audit.autosave.limit) 170
  specifying maximum size of internal log (options cifs.audit.logsize) 170
  specifying timestamp extensions (options cifs.audit.autosave.file.extension timestamp) 170
  updating (cifs audit save) 165
  valid size thresholds 167
  valid time intervals 168
  ways to save 165
file access auditing, enabling and disabling (options cifs.audit.file_access_events.enable) 163
file screening
  adding excluded extensions (fpolicy ext excl add) 154
  disabling (fpolicy disable) 152
  disconnecting file screening server (fpolicy servers stop default) 157
  displaying list of excluded extensions (fpolicy ext excl show) 154
  displaying screening server status (fpolicy servers show default) 157
  enabling (fpolicy enable) 151
  policy information, example 153
  removing excluded extensions (fpolicy ext excl remove) 155
  replacing list of excluded extensions (fpolicy ext excl set) 156
  requiring screening for file access (fpolicy options default required) 156
  resetting default file screening (fpolicy ext excl reset) 155

specifying screening 153
file screening policy 150
filer description, displaying and changing 66
home directories
accessing 127
creating directories in 125, 126
description of 115
naming styles, description of 117
paths, specifying (options cifs.home_dir) 120
specifying name style (options cifs.home_dir_namestyle) 124
home shares
stop offering (options cifs.home_dir) 131
syntax using UNC name 127
idle sessions
description of 183
timeout, specifying (options cifs.idle_timeout) 183
local groups
creating 110
definition of 110
deleting 112
removing members from 112
with SnapMirror 110
managing through Server Manager and User Manager 65
mapping inconsistencies with WAFL credential cache 322
mapping NT account to UNIX root 87
method of changing share-level ACL 135
migrating accounts to passwd/group files 91
multiprotocol to NTFS-only, effects of 193
NetBIOS aliases, creating (options cifs.netbios_aliases) 68
NetBIOS over TCP
description of 198
disabling (options cifs.netbios_over_tcp.enable) 198
NTFS-only to multiprotocol, effects of 192
oplocks
changing delay time for sending breaks (options cifs.oplocks.opendelta) 180
delay time for sending breaks, description 179
description of 177
filer and client oplock settings 178
managing 179
setting of for each qtree (qtree oplocks) 179
system-wide oplock (options cifs.oplocks.enable) 179
when to use 177
optimizing client access to NFS directory 312
passwd file when created 89
preventing from starting at reboot 188
quotas and generic user account 92
reconfiguring
cifs setup 193
description of 191
removing entries from ACL
from command line (cifs access) 140
with Server Manager 138
restarting
for a filer (cifs restart) 190
sending messages
description of 158
from command line (cifs broadcast) 159
from Server Manager 158
Server Manager, limitations of 105
setting specific users to receive UNIX credentials 73
shared directories, description of 95
share-level ACLs 97
reason for changing 135
using GIDs with 133
shares
creating using command line (cifs shares) 101
creating using Server Manager 100
deleting from command line (cifs shares) 104
deleting from Server Manager 104
description of 96
differences between home and other shares 115
displaying and changing properties from command line (cifs shares) 103

displaying and changing properties from Server Manager 102
forcegroup 96
GID, specifying effect on file access (options cifs.perm_check_user_gid) 134
group membership 96
number of users, specifying 96
shutdown messages, specifying users that receive 189
SID-to-name map cache
changing lifetime of entries (options cifs.sidcache.lifetime) 147
clearing (cifs sidcache clear) 148
description of 144
enabling and disabling (options cifs.sidcache.enable) 146
symbolic links
how CIFS clients follow 300
options for accessing 300
symbolic links as home directory names 117
tracing CIFS logins by NFS clients (options cifs.trace_login) 323
turning file policies on or off (options fpolicy.enable) 151
User Manager, limitations of 105
viewing event audit log (Event Viewer) 171
WAFL mapping inconsistencies 321
Windows guest accounts
creating/deleting 94
description of 94
configuration files, accessing 7
contents
of /etc/exports file 29
of /etc/passwd file 89
of /etc/usermap.cfg file 77, 78, 79
cyclic directory structures
description of 311

D

DAFS
client-server implementation 248
information required for setup 249
multiprotocol access to files 248
DAFS authentication
types of authentication sessions 261
types of client access 260
DAFS commands
dafs access (manages client access to DAFS exports) 270
dafs attach (attaches DAFS to a VI NIC) 258
dafs detach (detaches DAFS from a VI NIC) 259
dafs export (manages DAFS export points) 265
dafs setup (reconfigures DAFS) 250
dafs start (starts DAFS on a filer) 253
dafs stats (obtains DAFS statistics) 275
dafs status (displays status for DAFS protocol, sessions, and VI NICs) 280
dafs stop (stops DAFS on a filer) 253
DAFS exports
adding clients to the export's ACL 271
creating 266
destroying 267
disabling 269
displaying ACLs 274
enabling a disabled export 268
listing export points 267
managing access control lists (ACLs) for clients 270
modifying read or write attributes 268
reexporting 269
DAFS performance
displaying DAFS state 281
displaying DAFSsessions and requests 285
displaying filer statistics 276
displaying VI NIC sessions 284
displaying VI NIC status 282
obtaining DAFS statistics 277
viewing client sessions 286
DAFS protocol
persistently enabling or disabling 255
starting DAFS on a filer 254
stopping DAFS on a filer 254
transport requirements 248
DAFS VI NICs
attaching 257
detaching 259
data access management 15

Data ONTAP
converting to Unicode-formatted directories 313
creating Unicode-formatted directories (options wafl.create_ucode) 312
cyclic directory structures 311
directory structures, NFS and CIFS 312
file locking 296
file name creation 295
how usermap.cfg file is used 76
mapping NT names and UNIX names in 72
matching users to directories using CIFS 115
obtaining the GID 73, 89
obtaining the UID 72, 89
overview 12
read-only bit 297
SecureShare Access, description of 324
shared file permissions 297
using NTFS to determine file access 315
using UNIX credentials for filer access 72
directory structures
converting to Unicode-formatted directories 313
creating Unicode-formatted directories 312
cyclic directories 311
NFS and CIFS 312
disk shelves 11
disks
data 13
parity 13
spare 13
domain controller, selecting 195

E

environmental adapter, filer main unit 10
exported resources
access restrictions for 26
from different directory hierarchy levels 25
matched against NFS mount requests 24
rules for exporting 25
to a subnet, reasons for 41
ways to export 24
exporting resources, to a netgroup 40

F

file access
effects of GID in 134
mapping entries to WAFL credential cache 316
NFS access of CIFS files, description of 315
file locking
description of 296
NFS operations affected by 296
file name conventions, in multiprotocol environment 294
file names
conventions, in multiprotocol environment 294
creating lowercase (options cifs.save_case) 295
creation 295
file permissions, umask for (PC)NFS clients 60
file renaming
forcing lowercase (options cifs.save_case) 314
preventing 314
file screening policy 150
File Transfer Protocol (FTP). *See* FTP service
filer access
from a CIFS client 72
filer main unit components 9
filers
CIFS description 66
CIFS limitations with 105
managing CIFS through Server Manager and User Manager 65
restarting CIFS 190
FilerView, use for filer administration 8
FTP commands
options ftpd.anonymous.enable (enables or disables anonymous FTP access) 202
options ftpd.anonymous.home_dir (specifies a home directory for the anonymous user) 202
options ftpd.anonymous.name (specifies a name for anonymous login) 202
options ftpd.auth_style (specifies authentication style) 200
options ftpd.enable (enables the FTP daemon) 200

options ftpd.idle_timeout (specifies the timeout value for idle connections) 203
options ftpd.max_connections (specifies maximum number of concurrent FTP connections) 204
options ftpd.tcp_window_size (specifies the TCP window size) 204
FTP service
/etc/ftpusers. *See* Symbols
anonymous access
configuring 210
options for 202
authentication styles
about configuring 207
configuring Mixed 209
configuring NTLM 209
configuring UNIX 208
Mixed, description of 201
NTLM, description of 201
types of 200
UNIX, description of 200
denying users access to 203, 211
enabling 206
ftpd daemon 200
log files
types of 212
viewing 212
setting up 205
ftp.cmd file, description of 212
ftp.xfer file, description of 212

G

generic UNIX account
definition of 92
prerequisites for 92
generic user account
creating (options wafl.default_unix_user) 93
group membership, how determined by pcnfsd 57
guest accounts, Windows 94
guidelines for mapping NT/UNIX names 85

H

home directories
accessing others' home directories 128
CIFS accessing of 127
CIFS naming style in 117
CIFS, description of 115
creating CIFS directories 125, 126
naming style, specifying 124
wide symbolic links in 130
home directory paths
specifying (options cifs.home_dir) 120
specifying in /etc/home_dir file 122
home shares, stop offering for use (options cifs.home_dir) 131
HTTP commands
fail (adds fail rule to translation file) 223
httpstat (displays statistics) 240
httpstat (resetting statistics) 241
map (adds map to translation file) 217
options httpd.access (restricts access to HTTP services) 227
options httpd.admin.access (restricts access to filer administration via HTTP) 227
pass (adds pass rule to translation file) 221
redirect (adds redirect rule to translation file) 219
HTTP service
/etc/httpd.hostprefixes. *See* Symbols
/etc/httpd.mimetypes. *See* Symbols
/etc/httpd.passwd. *See* Symbols
/etc/httpd.translations. *See* Symbols
/etc/log/httpd.log. *See* Symbols
about setting up 214
license requirements 213
log file
information in 242
viewing 242
mapping file name suffix to MIME value 224
MIME Content-Type values 224
rules
fail, adding 223
map, adding 217
pass, adding 221
processing URL translation 216
redirect, adding 219
security
authentication methods 229

basic authentication 229
levels of 226
marking an untrusted subnet 228
NTLM authentication 230
protection, Web page 229
restricting access (ifconfig) 228
restricting access, options for 227
user password 229
using virtual firewall 228
Web page protection 229
starting the service 214
statistics
description of 238
displaying (httpstat) 240
resetting (httpstat) 241
testing the service 215
virtual hosting
description of 236
mapping virtual host addresses (ifconfig) 237
setting up 236

I

idle sessions
description of 183
timeout, specifying (options cifs.idle_timeout) 183

K

Kerberos, secure NFS access 47

L

LCD, filer main unit 10
licenses, HTTP service 213
local user accounts
managing 109
reasons for specifying 108
LUNs, defined 13

M

Map entries
creating 308
description 303
how to list 310
how used 303
requirements 303
mapped NT/UNIX names 76
mapping NT/UNIX names, guidelines for 85
Migrator, SecureShare Account 91
MIME Content-Type values 224
mount requests, matched with exported resources 24
multiprotocol environment, file naming conventions 294
multiprotocol filer to NTFS-only filer, effects of 193

N

name creation, file 295
names (usermap.cfg file), mapped NT/UNIX 76
names, guidelines for mapping NT/UNIX 85
NetBIOS aliases
/etc/cifs_nbalias.cfg file, about 69
creating in /etc/cifs_nbalias.cfg file 69
displaying list of (cifs nbalias) 69
processing entries in cifs_nbalias.cfg file (cifs nabalias load) 69
ways to create 68
NetBIOS over TCP
description of 198
disabling 198
NetBIOS, aliases 68
network commands
ifconfig (mapping virtual host addresses) 237
ifconfig (restricting HTTP access over subnet) 228
NFS commands
exportfs (exports entries in exports file) 44
exportsfs (exports all entries in exports file) 29
option nfs.v4.id_domain (specifies user ID domain) 56
options nfs.kerberos.file_keytab.enable (enables file-based keytab) 52
options nfs.kerberos.principal (sets up an NFS server principal) 52
options nfs.kerberos.realm (specifies the

Kerberos realm) 52
options nfs.vN.enable (enables and disables NFS versions) 53
options pcnfsd (enables and disables pcnfsd) 58
options pcnfsd.umask (defines umask) 60
NFS protocol
(PC)NFS
clients, authentication of 57
creating user entries with 59
absolute symbolic links, redirecting for CIFS clients 308
accessing CIFS files, description of 315
contents of exports file 29
different restrictions for different hosts, specifying 31, 32, 33
enabling and disabling versions (options nfs.vN.enable) 53
exportfs command
purposes of 43
rules for specifying multiple access restrictions 43
targets that can be specified 43
exporting
entries differently than specified in exports file 44
resources to a netgroup 40
resources to subnet 42
to netgroups, restrictions of 39
to targets 29
file renaming by CIFS clients, prevention of 314
format of export target to subnet 41
format of netgroup entry 39
mapping inconsistencies with WAFL credential cache 321
mount requests, matched with exported resources 24
netgroups, description of 39
optimizing CIFS access to NFS directory 312
pcnfsd daemon
determining group membership 57
enabling and disabling (options pcnfsd) 58
relationship of netgroup and netgroup.byhost map 39
resources clients can mount 24
restrictions for all hosts, specifying 31
root access restrictions for all clients, specifying 34
root access restrictions for some clients, specifying 35, 36
root access with 34
rules for specifying an exports entry 30
same root access restrictions for all clients, specifying 35
secure NFS access 47
secure NFS access, implementation restrictions of 47
secure NFS access, setting 48
security service
description of 37
specifying 37
symbolic links, description of with CIFS 299
umask
and (PC)NFS-created files 60
definition of (options pcnfsd.umask) 60
unexporting resources 45
version 4
limitations 54
pseudo-fs effect on mount points 55
specifying user ID domain (option nfs.v4.id_domain) 56
support 54
WAFL mapping inconsistencies 321
WebNFS
configuring 61
NT/UNIX names (usermap.cfg file), mapped 72, 76
NT/UNIX names, guidelines for mapping 85
NTFS-only filer to multiprotocol filer, effects of 192
NVRAM 9

O

oplocks
changing delay time for sending breaks (options cifs.oplocks.opendelta) 180
delay time for sending breaks, description 179
description of 177
filer and client settings 178

managing 179
setting of for each qtree (qtree oplocks) 179
system-wide (options cifs.oplocks.enable) 179
when to use 177
options
ftpd.anonymous.name 202

P

pcnfsd daemon
determining group membership 57
enabling and disabling (options pcnfsd) 58
how it provides authentication 57
permissions
changing UNIX, description of 325
changing, using SecureShare Access 325
for shared files 297
preferred domain controller, specifying 195

Q

qtrees
administrative actions 14
description 14
setting oplocks for each (qtree oplocks) 179

R

RAID groups, definition of 12
read-only bit
deleting files when set (options cifs.perm_check_ro_del_ok) 298
description of 297
resources
definition of 23
NFS clients can mount 24
unexporting of 45
ways to export 24
root directory, for WebNFS lookup 61
rules
for exporting resources 25
URL translation 216

S

SecureShare Access
changing permissions with 325
description of 324
installing 324
SecureShare Account Migrator 91
security
levels for HTTP 226
restricting HTTP access 227
security styles
Kerberos 37, 47
client support 47
enabling keytab (nfs.kerberos.file_keytab.enable) 52
implementation 47
setting up an NFS server principal (options nfs.kerberos.principal) 52
setting up for UNIX 49
setting up for Windows 2000 48
specifying the Kerberos realm (options nfs.kerberos.realm) 52
specifying for NFS access 37
serial ports, filer main unit 10
Server Manager
adding entries to ACL 136
changing CIFS filer description 66
creating a share 100
disconnecting clients 185
displaying and changing ACLs 135
displaying share properties 102
limitations of 105
managing CIFS 65
removing entries from ACL 138
sending messages 158
session information (CIFS), displaying 181
shared directories, description of 95
shared resources
with NFS and (PC)NFS clients 23
with WebNFS clients 23
share-level ACLs
changing 135
using GIDs 133
shares
creating
from command line (cifs shares) 101

from Server Manager 100
deleting 104
description of 96
differences between home and other shares 115
displaying and changing properties 102, 103
from command line (cifs shares) 103
forcegroup 96
group membership 96
number of users, specifying 96
share boundary checking for symbolic links
disabling 306
specifying 97
umask value, description 98
virus scanning, specifying 99
wide symbolic links, enabling 98
SID-to-name map cache
changing lifetime of entries 147
clearing (cifs sidcache clear) 148
description of 144
enabling and disabling 146
slots, filer main unit 10
statistics
HTTP, description of 238
symbolic links
absolute links
redirecting for CIFS clients 308
redirecting using Map entries 303
redirecting using Widelink entries 304
as CIFS home directory names 117
description of 299
disabling share boundary checking for
description of 306
limitations of 306
requirements for 306
enabling CIFS clients to follow (options cifs.symlinks.enable) 300
options for CIFS access to 300
preventing links to files 301
ways to redirect 308
wide, enabling for a share 98
system board, filer main unit 9
system memory, filer main unit 9

T

translation, rules for URL 216

U

umask value, description 98
UNC name, syntax for specifying home share 127
Unicode format, directory structure 312
UNIX credentials, description of 72
UNIX permissions, changing 325
UNIX user name, special characters in 70
URL translation rules 216
user authentication method, displaying 108
User Manager
creating a local group 110
deleting group 112
limitations of 105
managing CIFS 65
removing members from group 112

V

virtual firewall, for HTTP 228
virtual hosting for HTTP 236
virus scanning, specifying for a share 99
volumes
administrative actions 14
defined 13
RAID groups, definition of 12

W

WAFL credential cache
adding entries to (wcc) 316
deleting entries in (wcc) 317
description of 315
displaying cache statistics (wcc) 319
mapping inconsistencies
description of 321
with a CIFS name 322
with a UNIX name 321
monitoring 319
setting length of validity (options wafl.wcc_minutes_valid) 318
Web page protection 229

WebNFS
configuring 61
root directory 61
wide symbolic links
enabling for a share 98
in home directories 130
Widelink entries
creating 309
description 304
how to list 310
how used 304
limitations 304
requirements 304
Windows guest account 94
Windows NT commands, performing Windows filer tasks 7
Windows user name
conversion from Windows 2000 format to pre–Windows 2000 format 70
different formats 70, 81
WINS server addresses, changing 191