Data ONTAP™ 7.0.1
File Access and Protocols 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 210-01115-A0
April 2005

Copyright and trademark information

Copyright information

Copyright © 1994–2005 Network Appliance, Inc. All rights reserved. Printed in the U.S.A.

No part of this document 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.

This software contains materials from third parties licensed to Network Appliance Inc. which is sublicensed, and not sold, and title to such material is not passed to the end user. All rights reserved by the licensors. You shall not sublicense or permit timesharing, rental, facility management or service bureau usage of the Software.

Portions developed by the Apache Software Foundation (http://www.apache.org/). Copyright © 1999 The Apache Software Foundation.

Portions Copyright © 1995–1998, Jean-loup Gailly and Mark Adler

Portions Copyright © 2001, Sitraka Inc.

Portions Copyright © 2001, iAnywhere Solutions

Portions Copyright © 2001, i-net software GmbH

Portions of this product are derived from version 2.4.11 of the libxml2 library, which is copyrighted by the World Wide Web Consortium.

Network Appliance modified the libxml2 software on December 6, 2001, to enable it to compile cleanly on Windows, Solaris, and Linux. The changes have been sent to the maintainers of libxml2. The unmodified libxml2 software can be downloaded from http://www.xmlsoft.org/.

Copyright © 1994–2002 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. http://www.w3.org/Consortium/Legal/

Software derived from copyrighted material of the World Wide Web Consortium is subject to the following license and disclaimer:

Permission to use, copy, modify, and distribute this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications, that you make:

The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.

Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, a short notice of the following form (hypertext is preferred, text is permitted) should be used within the body of any redistributed or derivative code: "Copyright © [$date-of-software] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. http://www.w3.org/Consortium/Legal/.

Notice of any changes or modifications to the W3C files, including the date changes were made.

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

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 or purchase of this product does 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, the Network Appliance logo, the bolt design, NetApp–the Network Appliance Company, DataFabric, FAServer, FilerView, MultiStore, NearStore, NetCache, SecureShare, SnapManager, SnapMirror, SnapMover, SnapRestore, SnapVault, SyncMirror, and WAFL are registered trademarks of Network Appliance, Inc. in the United States, and/or other countries. Data ONTAP, gFiler, Network Appliance, SnapCopy, Snapshot, and The Evolution of Storage are trademarks of Network Appliance, Inc. in the United States and/or other countries and registered trademarks in some other countries. ApplianceWatch, BareMetal, Camera-to-Viewer, ComplianceClock, ComplianceJournal, ContentDirector, ContentFabric, EdgeFiler, FlexClone, FlexVol, FPolicy, HyperSAN, InfoFabric, LockVault, Manage ONTAP, NOW, NetApp on the Web, ONTAPI, RAID-DP, RoboCache, RoboFiler, SecureAdmin, Serving Data by Design, SharedStorage, Simulate ONTAP, Smart SAN, SnapCache, SnapDirector, SnapDrive, SnapFilter, SnapLock, SnapMigrator, SnapSuite, SnapValidator, SohoFiler, vFiler, VFM, Virtual File Manager, VPolicy, and Web Filer are trademarks of Network Appliance, Inc. in the United States and other countries. NetApp Availability Assurance and NetApp ProTech Expert are service marks of Network Appliance, Inc. in the United States. Spinnaker Networks, the Spinnaker Networks logo, SpinAccess, SpinCluster, SpinFS, SpinHA, SpinMove, and SpinServer are registered trademarks of Spinnaker Networks, LLC in the United States and/or other countries. SpinAV, SpinManager, SpinMirror, SpinRestore, SpinShot, and SpinStor are trademarks of Spinnaker Networks, LLC in the United States and/or other countries.

Apple is a registered trademark and QuickTime is a trademark of Apple Computer, Inc. in the United States and/or other countries. Microsoft is a registered trademark and Windows Media is a trademark of Microsoft Corporation in the United States and/or other countries. RealAudio, RealNetworks, RealPlayer, RealSystem, RealText, and RealVideo are registered trademarks and RealMedia, RealProxy, and SureStream are trademarks of RealNetworks, Inc. in the United States and/or other countries.

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

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

Network Appliance NetCache is certified RealSystem compatible.

Table of Contents

Preface xi

Chapter 1 **Introduction to File Access Management** 1

Chapter 2 **File Access Using NFS** 3

How resources become available to NFS clients 4

Access restrictions for exported resources 6

How /etc/exports works 9
- Specifying access restrictions in /etc/exports 12
- Specifying root access in /etc/exports 15
- Specifying security services 19
- Exporting resources to netgroups from /etc/exports 21
- Exporting resources to subnets from /etc/exports 23

Exporting or unexporting resources using the exportfs command 26

Providing secure NFS access 31
- NFS mountd traces 45

Enabling and disabling NFS versions 46

Using NFSv4 47

File sharing with (PC)NFS clients 54
- Enabling or disabling the pcnfsd daemon 55
- Creating (PC)NFS user entries on the filer 56

Defining the umask for (PC)NFS-created files and directories 57

Configuring Data ONTAP for WebNFS 58

Chapter 3 **File Access Using CIFS** 61

Configuring CIFS using cifs setup 63

File management through Windows administrative tools 67

Displaying and changing the description of the filer 68

Creating NetBIOS aliases for the filer 69

Disabling NetBIOS over TCP 71

How to specify a filer's Windows name 72

How to change a filer's computer account password. 74

How CIFS users obtain UNIX credentials 75
How Data ONTAP maps user names 79
How to specify entries for the /etc/usermap.cfg file 80
How Data ONTAP interprets domain names in /etc/usermap.cfg 84
Examples of usermap.cfg entries. 85
Guidelines and recommendations for mapping user names 88
Mapping a Windows account to root. 90
Mapping UNIX names to UIDs and GIDs 92
Migrating CIFS user accounts to the passwd and group files 94
Creating or disabling the default UNIX user account 95
Enabling or disabling the Windows guest user account 97

Sharing directories 99

Displaying and changing share properties 107

Deleting a share. 110

CIFS limitations 111

Understanding authentication issues 114

Using null sessions to access filer storage in non-Kerberos environments. . .116

Understanding local user accounts 119

Creating local groups on the filer. 121

Understanding home directories on the filer 126

How Data ONTAP matches a directory with a user 128

Setting up user home directories 130
Specifying home directory paths. 131
Specifying the naming style of home directories 134
Creating directories in a home directory path 135

Accessing home directories. 138

Specifying support for wide symbolic links in home directories. 141

How to stop offering home directories 142

How share-level access control lists work 143

Specifying how group IDs work with share-level ACLs. 144

Changing and displaying a share-level ACL 146

Changing and displaying file-level ACLs 153

Understanding the SID-to-name map cache 156
Enabling and disabling the SID-to-name map cache. 158

Changing the lifetime of SID-to-name mapping entries 159
Clearing the SID-to-name map cache 160

Sending a message to all users on a filer 162

Understanding CIFS auditing. 164
Configuring Data ONTAP for CIFS auditing 166
Saving and clearing audit events 168
Understanding event detail displays 176

Understanding oplocks 180

Managing oplocks 182

Displaying CIFS session information 184

Timing out idle sessions 186

Disabling CIFS service 187
Disconnecting selected clients 188
Disabling CIFS for the entire filer 190
Specifying which users receive CIFS shutdown messages 192

Restarting CIFS service. 193

Reconfiguring CIFS for the filer 194

Selecting domain controllers and LDAP servers 198

SMB Signing Support 203

Tracking Statistics 204

Chapter 4 **File Access Using FTP**. 207

Understanding FTP service on the filer. 208

Setting up FTP service on your filer 214
Enabling the FTP service on your filer. 215
Changing the FTP file size 216
FTP file locking 217
Configuring an authentication style 218
Configuring anonymous FTP access 221

Denying access to the FTP service 222

Viewing FTP log files and connection statistics 223

Number of log files per FTP session 225

Managing FTP connections. 226

Chapter 5 **File Access Using HTTP** 229

Starting the HTTP service 230

Testing the HTTP service. 231

Specifying the filer response to HTTP requests 232
Adding the map rule 233
Adding the redirect rule 235
Adding the pass rule 237
Adding the fail rule. 239

Configuring MIME Content-Type values 240

Maintaining HTTP service security 242
Using HTTP options to restrict access 243
Using an HTTP virtual firewall 244
Protecting Web pages 245
Editing the /etc/httpd.access file 247
Creating and editing httpd.passwd and httpd.group files 250

Using virtual hosting 252

Displaying HTTP statistics 254

Viewing HTTP connection information 258

Chapter 6 **File Access Using WebDAV**. 261

Understanding the WebDAV protocol 262

Chapter 7 **File Sharing Between NFS and CIFS** 265

Understanding NFS and CIFS file naming 266

Understanding file locking between protocols 268

Understanding read-only bits 269

Using LDAP services. 271

File screening using FPolicy 286

Enabling and disabling file screening 288

Managing file policies 291

Managing file screening servers 296

Controlling CIFS access to symbolic links 297
Understanding how CIFS clients interact with symbolic links 298
About Map entries 301

About Widelink entries 302
About disabling share boundary checking for symbolic links 304
Redirecting absolute symbolic links 306
Preventing access by CIFS clients to cyclic directory structures 309

Optimizing NFS directory access for CIFS clients 310

Preventing CIFS clients from creating uppercase file names 312

Accessing CIFS files from NFS clients. 313
Adding mapping entries to the WAFL credential cache 314
Deleting mapping entries from the WAFL credential cache 315
Setting how long mapping entries are valid 317
Monitoring WAFL credential cache statistics 318
Managing mapping inconsistencies 321

Installing SecureShare Access 325

Changing UNIX permissions and DOS attributes from Windows 326

Chapter 8 **File Access Using DAFS** 329

Understanding the DAFS protocol 330

Understanding the DAFS service implementation 332

Configuring the DAFS service 333

Starting and stopping the DAFS service 337

Attaching and detaching network devices 341

Understanding session authentication 344

Managing exports for DAFS service 349

Managing client host access to DAFS exports 354

Measuring DAFS performance 359

Displaying DAFS status 364

Configuring DAFS with options 373

Glossary 377

Index 387

Preface

Introduction

This guide describes how to configure, operate, and manage Network Appliance™ filers that run Data ONTAP™ 7.0.1 software. It includes all supported filer models.

Audience

This guide is for system administrators who are familiar with operating systems such as UNIX® and Windows®, 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), Hypertext Transport Protocol (HTTP), File Transport Protocol (FTP), Direct Access File System (DAFS) protocol, and Web-based Distributed Authoring and Versioning (WebDAV) 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

Network Appliance's storage products (filers, FAS appliances, and NearStore® systems) are all storage systems — also sometimes all called filers or 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.

FilerView as an alternative to commands

Tasks you perform as a Data ONTAP administrator can be performed by entering commands at the filer console, in configuration files, or through a Telnet session or a Remote Shell connection.

Another method of performing many common tasks is to use the FilerView® graphical management interface for viewing and managing a filer from a Web browser. FilerView comes with every filer, is easy to use, and includes Help that explains Data ONTAP features and how to work with them in FilerView.

For more information about accessing a filer with FilerView, and about FilerView Help, see the *System Administration Guide*.

Command conventions

In examples that illustrate commands executed on a UNIX workstation, 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 File Access Management 1

About this chapter

This chapter describes how to publish files that can be accessed and modified by clients running different protocols, Data ONTAP features that allow you to manage and control access to those files.

Supported protocols

Data ONTAP provides an infrastructure to manage files (resources) and the accounts of users trying to access those files. This infrastructure includes the mapping of read and write permissions for users and groups, regardless of the protocol being used by the file creator and the user trying to access that file.

Data ONTAP maps these permissions across the following protocols:

- NFS
- CIFS
- FTP
- HTTP
- WebDAV
- DAFS

Factors used to control file access

Authentication: In addition to enforcing file-based permissions, Data ONTAP also enforces authentication-based access restrictions. Authentication restrictions determine whether a user can connect to the storage appliance at all. Data ONTAP supports Kerberos authentication from both UNIX and Windows servers. Using authentication, Data ONTAP can control file access by denying access to an entire filer or vFiler™ unit, based upon which user is trying to connect and which client machine they are using to make that connection.

The remaining chapters of this guide describe procedures and processes that are specific to each protocol, for the purpose of providing, controlling, and monitoring file access.

File permission attributes: When a file is created, a list of access permissions is generated. While the form of that permissions list varies with each protocol, Data ONTAP maps common permissions, such as reading and writing to the file. When a user tries to access a file, Data ONTAP uses the permissions

list to determine whether to permit access. Access can be permitted or denied based upon the operation being performed (such as reading or writing), and upon the following factors:

- user account
- user group or netgroup
- client protocol
- client IP address
- file type

Using the lookup service you specify—Lightweight Directory Access Protocol (LDAP), Network Information Service (NIS), or local filer information—Data ONTAP determines who is trying to access the resource, and verifies that the permission list indicates operations that can be performed on that resource, by that user.

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 4
- "Access restrictions for exported resources" on page 6
- "How /etc/exports works" on page 9
- "Exporting or unexporting resources using the exportfs command" on page 26
- "Providing secure NFS access" on page 31
- "Enabling and disabling NFS versions" on page 46
- "Using NFSv4" on page 47
- "File sharing with (PC)NFS clients" on page 54
- "Defining the umask for (PC)NFS-created files and directories" on page 57
- "Configuring Data ONTAP for WebNFS" on page 58

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 from the command line to export the resources according to the specifications in the /etc/exports file.

  This method exports resources that you want to make available automatically, every time the filer reboots.

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

  This method exports resources temporarily; 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 9. For information about how to use the `exportfs` command, see “Exporting or unexporting resources using the exportfs command” on page 26.

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, because it is not a path name to a file, directory, or volume. 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, unlike SunOS, Data ONTAP permits 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 -rw=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 using export options in the /etc/exports file or with command line options for the `exportfs` command. (These options are identical using either method.) Access restriction 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
- Whether files can be created with the SETUID bit
- The user ID (UID) of the user accessing the resource

When multiple restrictions are applied to the same resource, the most restrictive option always takes precedence.

Meanings of access restriction options

The root option: This option 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.

The rw option: It gives read-write access to the specified hosts. If no hosts are specified, all hosts have read-write access.

The ro option: It gives read-only access to the specified hosts. If no hosts are specified, all hosts have read-only access.

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.
- If a name is provided instead of a UID, that name is looked up according to the order specified in the /etc/nsswitch.conf file to determine the corresponding UID to be assigned by the `anon` option.

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 applicable access restrictions.

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

```
/vol/vol0 -ro=host1,root=host1,rw=host2
```

The `ro` 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. Because the most restrictive option takes precedence, 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 -ro=host1:host2,root=host1
```

The `ro` 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 the list of resources to be exported when Data ONTAP starts up, restarts, or receives the `exportfs -a` command (given that a valid NFS license exists). Maintaining the /etc/exports file enables resources to be available automatically upon filer startup and eliminates the need to enter the `exportfs` command for each resource you want to export.

In the /etc/exports file, you also specify the access restrictions and security service with which clients can mount each resource. At any time after filer startup, you can override specifications in the /etc/exports file by using the `exportfs` command (until the filer is restarted or another overriding `exportfs` command is issued).

Automatic changes to /etc/exports

Data ONTAP automatically updates the exports file when you enter the following Data ONTAP commands:

- `vol create`
- `vol destroy`
- `vol rename`
- `exportfs -p`

Volume create entries:

New volumes you create will have /etc/exports entries with the following default assignments.

When no administrative host has been specified, Data ONTAP adds an /etc/exports entry with read and write permissions on that volume.

```
/vol/volX  -rw,nosuid
```

When you have defined an administrative host using the `setup` command, Data ONTAP adds an /etc/exports entry that is associated with the administrative host and has read and write permissions on that volume.

```
/vol/volX  -rw,root=ADMIN,nosuid
```

When changes to /etc/exports take effect

When entries in the /etc/exports file are added or changed, either by the administrator or automatically, those changes take effect when one of the following events occurs:

- Data ONTAP executes the `exportfs -a` command. This exports all entries in the /etc/exports file.
- Data ONTAP executes the `exportfs -r` command. This re-exports all entries in the /etc/exports file into memory and removes from memory all non-persistent (command line) exports.
- 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 26.

Targets to which you export a resource

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

- DNS subdomains
- 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 by all hosts as a read-write resource.
- If an /etc/exports file entry contains the path name of a resource followed by export options, the resource is exported with the specified 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 (,). Multiple hosts associated with the same access restriction are separated by a colon (:).

Example: `/vol/vol0 -ro=host1:host3,root=host1,rw=host2`

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

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 12
- "Specifying root access in /etc/exports" on page 15
- "Specifying security services" on page 19
- "Exporting resources to netgroups from /etc/exports" on page 21
- "Exporting resources to subnets from /etc/exports" on page 23

Specifying access restrictions in /etc/exports

How to specify access restrictions

You specify access restrictions by using a text editor to modify the /etc/exports file. Access restrictions take effect only after you subsequently use the `exportfs` command.

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	In the /etc/exports file, 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	In the /etc/exports file, 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 specified hosts to mount read-write: To allow only certain hosts to mount the resource read-write and to prevent other hosts from mounting the resource, complete the following step.

Step	Action
1	In the /etc/exports file, enter the path name of the resource with the `rw` option. **Example:** `/vol/vol0/home -rw=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 specified hosts to mount read-only: To allow certain hosts to mount the resource read-only and prevent other hosts from mounting the resource, complete the following step.

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

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

Step	Action
1	In the /etc/exports file, enter the path name of the resource with the `ro` and `rw` options. **Example:** `/vol/vol0/home -ro,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 specified hosts to mount read-write and specified hosts to mount read-only: To allow certain hosts to mount a resource read-write and certain other hosts to mount the resource read-only, complete the following step.

Step	Action
1	In the /etc/exports file, enter the path name of the resource with the `rw` and `rw` options. **Example:** `/vol/vol0/home -ro=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. Other hosts 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 has one of the following types of access:

- The root users is able to access the resource only with the UID 0.
- The root users is able to access the resource with a nonroot UID.
- The root users is 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	In the /etc/exports file, enter the path name of the resource with the `anon` option. **Example:** `/vol/vol0/home -anon=0` **Result:** Root users on any client can access the /vol/vol0/home directory with UID 0.

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

Step	Action
1	In the /etc/exports file, enter the path name of the resource with the `anon` option. **Example:** `/vol/vol0 -anon=root` **Result:** Users on any client can access the /vol/vol0 directory.

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

Step	Action
1	In the /etc/exports file, 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	In the /etc/exports file, 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	In the /etc/exports file, 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 specified clients to have root access and all others no root access: To have the root users on certain clients retain the root UID and to give all others no root access, complete the following step.

Step	Action
1	In the /etc/exports file, 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 all other clients are denied access.

For root users on specified clients to have root access and all others nonroot access: To have users on certain 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 specified clients to have root access and others to be considered to be “nobody”: To have root users on certain clients retain the root UID and root users on other clients be considered the “nobody” user, complete the following step.

Step	Action
1	In the /etc/exports file, 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 31.

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, from least to most secure, are:

- UNIX (AUTH_SYS) authentication—Does not use strong cryptography and is the least secure of the security services. This is the default security service used by Data ONTAP.
- Kerberos version 5 (v5) — Provides three methods of security:
  - Authentication (krb5) — Uses strong cryptography to prove a user's identity to a filer and to prove a filer's identity to a user.
  - Integrity (krb5i) — Provides a cryptographic checksum of the data portion of each request and the response message to each request. This defends against "man in the middle" tampering with filer NFS traffic.
  - Privacy (krb5p) — Encrypts the contents of packets bidirectionally, including procedure arguments and user data, using a shared session key established by the client from the filer.

Note

Before you can specify a Kerberos v5 security setup, Kerberos services must be set up for NFS. For more information about setting up NFS, see "Providing secure NFS access" on page 31.

Specifying security services

Data ONTAP allows you to specify more than one security service. This enables users with different types of security to have NFS access. If you want to require 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 must use 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 your 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 whenever the NIS server is not available.

Restrictions of the /etc/netgroup file

Each line in the /etc/netgroup file can contain up to 4,096 characters. Make sure you stay within this constraint when adding entries to this file.

Format of a netgroup entry

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

- Another netgroup name
- An entry in the form (*host_name*, *user_name*, *domain_name*)
  - *host_name* is a fully qualified name, if the host is not in the local domain.
  - *user_name* is ignored. (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.

Note

Even if you leave a field empty, you must include the comma to keep the entry syntax correct.

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 `rw`, `ro`, and `root` options.

Exporting a resource to a netgroup

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

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="3">1</td><td>If...</td><td>Then...</td></tr>
<tr><td>You want the filer to use NIS for authenticating (PC)NFS users</td><td>Go to Step 2.</td></tr>
<tr><td>The filer is not an NIS client or you want to use different netgroup definitions from the ones specified in the netgroup.byhost map</td><td>Create the /etc/netgroup file, then go to Step 2.</td></tr>
<tr><td>2</td><td colspan="2">Type the following entry in the /etc/exports file:
path_name -rw=netgroup_name[:netgroup_name:...]
Example: The /etc/exports file contains the following entries:
<code>/vol/vol0 -root=trusted-hosts,rw=trusted-hosts</code>
<code>/vol/vol0/home -rw=untrusted-hosts</code>
Result: Hosts that belong to the trusted-hosts netgroup can mount the /vol/vol0 volume as read-write. The root user on any host which belongs to the trusted-hosts netgroup 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.</td></tr>
</table>

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 enter 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 on the subnet mask.
	Examples: If you want to export to IP addresses of the form 10.45.67.x with 24 leading 1 bits in the subnet mask, specify the following export target:
	`10.45.67.0/24`
	If you want to export to IP addresses of the form 10.45.x.y with 16 leading 1 bits in the subnet mask, specify the following export target:
	`10.45.0.0/16`

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.
	
	Example: If you want to export to IP addresses of the form 10.45.67.x with a netmask of 255.255.255.0, use these values as the export target:
	
	`10.45.67 255.255.255.0`
	
	`network 10.45.67 netmask 255.255.255.0`

Network address and subnet mask for multiple subnets	**Example:** If you want to export to subnets 10.45.67 and 10.45.68 with a netmask of 255.255.255.0, you could export them with the following entries:
	
	`network 10.45.67.0 netmask 255.255.255.0`
	`network 10.45.68.0 netmask 255.255.255.0`
	
	Note that all four IP octets are required when exporting to subnets using this format.
	
	Example: The prior example could also be exported using the following syntax:
	
	`10.45.67/24`
	`123.45.68/24`
	
	Note that all four IP octets are *not* required for /etc/exports entries that use this format. You can use both formats shown in the examples above within the same exports file.

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* -ro=*subnet_address*[*:subnet_address*:...]** **Example:** `/vol/vol0/home -ro=10.45.67/24` **Result:** Hosts with IP addresses in the form of 10.45.67.x can mount the /vol/vol0/home directory as read-only. **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 12 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 to

- Export entries specified in the /etc/exports file
- Export entries not specified in the /etc/exports file
- Modify restrictions on resources already specified in the /etc/exports file
- Unexport entries that are currently exported. Unexported, resources become unavailable to clients
- Determine the actual (absolute) storage path to an exported resource
- Flush the access cache
- Check the access permissions of a host for an export entry

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 ro=host1:host3,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 21 for information about how to define a netgroup
- Subnets. See “Exporting resources to subnets from /etc/exports” on page 23 for information about the format of a subnet address
- DNS subdomains

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.

Pre-caching entries in /etc/exports

To determine the pre-caching of entries in the /etc/exports file, complete the following step.

Step	Action
1	Enter the following command: **`exportfs -c host path_name [ro|rw|root]`** *host* is the dotted IP address of the client to check in the access cache. *path_name* is the path name to a resource specified in the /etc/exports file. `ro` | `rw` | `root` checks for read-only (ro), read-write (rw), or root permissions. If this parameter is not provided, the `exportfs -c` command checks to see if the client has mount permissions to export.

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.
	Note
	When you do not specify the **`-i`** option on the command line, all command line options for the export will be ignored for resources that have different options specified in the /etc/exports file. The options from the corresponding entry in the /etc/exports file will be used instead.
	options specifies the access restrictions. These can be `access`, `ro`, `rw`, or `anon`.
	path_name is the path name to the resource you want to export.
	Examples: The following `exportfs` command exports the /vol/vol1 volume, giving read-only access to a netgroup named trusted_hosts, giving root access to host1 and giving read-write nonroot access to host2:
	`exportfs -i -o ro=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 ro=untrusted_hosts /vol/vol2`

Unexporting resources

To unexport resources, complete the following steps.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="3">1</td><td>If...</td><td>Then...</td></tr>
<tr><td>You want to unexport all resources that are currently exported</td><td>Go to Step 2.</td></tr>
<tr><td>You want to unexport a specific resource that is currently exported</td><td>Go to Step 3.</td></tr>
<tr><td>2</td><td colspan="2">Enter the following command:
<tt>exportfs -au</tt>
Result: All currently exported resources are unexported immediately.</td></tr>
<tr><td>3</td><td colspan="2">Enter the following command:
<tt>exportfs -u path_name</tt>
path_name is the path name to the resource that you want to unexport.
Result: The specified resource is unexported immediately.</td></tr>
</table>

Providing secure NFS access

What secure NFS access does

Secure NFS access uses an authentication protocol to ensure the security of data and the 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 two Kerberos Key Distribution Center (KDC) types:

KDC Type	Description
Active Directory-based	The Windows domain is the Kerberos realm. CIFS and the Active Directory server must use the same domain controller.
UNIX-based KDC	The UNIX domain is not always equivalent to the Kerberos realm. Because of the unavailability of UNIX-based KDCs that support Kerberos for CIFS, a filer configured to run both an NFS and CIFS server cannot support Kerberos for CIFS.

Note

A filer can support only one KDC type at a time.

NFS clients supporting Kerberos v5 security

Kerberos v5 authentication supports the following popular clients:

Note

The list of clients that support Kerberos v5 security only includes widely used clients that have been tested either in Network Appliance's laboratory or at interoperability test events, such as Connectathon (www.connectathon.org). For information about using Kerberos v5 with other clients, contact your NFS client vendor and verify that their products support RFC1964 and RFC2203.

Operating system	Versions supported for Kerberos v5	Required software and availability notes
Linux	Linux 2.6 running NFS version 4	No additional software is necessary.
Solaris	Solaris 2.6 running NFS version 2 (NFSv2) or NFS version 3 (NFSv3)	Sun Enterprise Authentication Mechanism (SEAM) 1.0, available in Sun Microsystems' Solaris Easy Access Server (SEAS) 3.0 product bundle
	Solaris 7 running NFSv2 or NFSv3	SEAM 1.0, available from Sun Microsystems' SEAS 3.0 product bundle
	Solaris 8 running NFSv2 or NFSv3	SEAM 1.0.1, available from Sun Microsystems' Solaris 8 Admin Pack or the Solaris 8 Encryption Pack at www.sun.com
	Solaris 9 running NFSv2 or NFSv3	No additional software is necessary.
	Solaris 10 running NFSv2, NFSv3, or NFSv4	No additional software is necessary.
Windows	Windows clients running NFSv2 or NFSv3	Hummingbird NFS Maestro version 7 or NFS Maestro Solo version 7
	Windows clients running NFSv2, NFSv3, or NFSv4	Hummingbird NFS Maestro Client version 8 or NFS Maestro Solo version 8

Specifying secure NFS

You specify secure NFS using the `exportfs` command or by setting the resource entry in the /etc/exports file. For information about exporting resources with security services, see "Specifying security services" on page 19.

To specify secure NFS, complete the following steps.

Step	Action
1	Edit the resource entry in the /etc/exports file to include the security service you want. **Example:** `/vol/vol0 -rw=host1,sec=krb5`
2	Enter the following command: **`exportfs`** ***`path_name`*** *path_name* is a file or directory. **Example:** `exportfs /vol/vol0`

Enabling Kerberos v5 security services for NFS

Before the security options you specify using the `exportfs sec` option will work, you need to use `nfs setup` to enable security services. The `nfs setup` script prompts you to specify either an Active Directory Kerberos v5 Key Distribution Center (KDC), or a UNIX-based KDC. Steps for configuring the filer to use an Active Directory KDC and a UNIX-based KDC are described separately, in the following sections.

Setting up security services using a Windows Active Directory-based KDC

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

There are two scenarios for using `nfs setup` to configure Data ONTAP for an Active Directory-based KDC:

- If have already run `cifs setup` and configured Data ONTAP to use Active Directory for CIFS, `nfs setup` automatically uses some of the configuration information you specified for CIFS.
- If have not run `cifs setup` to configure CIFS, you need to enter the configuration information that would otherwise have been taken from your CIFS configuration.

Note

Regardless of which protocol you configure first, you must configure CIFS and NFS Kerberos v5 security to use the same Active Directory realm.

Setting up NFS Active Directory KDC services before configuring CIFS

To set up Kerberos security services using a Windows Active Directory KDC before CIFS is set up, you need to do the following:

- Configure the filer to use Active Directory as the DNS server.
- Configure the filer to use your Active Directory KDC.

Configuring the filer to use Active Directory as the DNS server:

To configure the filer to use Active Directory-based domain name service, modify the /etc/resolv.conf file as necessary to ensure that it lists only Active Directory servers, as described in the following step.

Step	Action

1	Modify the filer's /etc/resolv.conf file as necessary to list only the Active Directory servers for the realm.
	Example:
	For a Kerberos realm in which the Active Directory servers are 172.16.1.180 and 172.16.1.181, change /etc/resolv.conf to include only the following Active Directory server entries:
	`nameserver 172.16.1.180`
	`nameserver 172.16.1.181`
	Make sure you remove all other Active Directory server entries for that realm.

Configuring the filer to use an Active Directory-based KDC:

To configure the filer to use an Active Directory-based KDC, run the `nfs setup` script and enter configuration information as described in the following steps.

Note

If you have already used `nfs setup` to enter configuration information, the prompts you receive may differ from those shown in the following procedure.

Step	Action
1	Enter the following command: **`nfs setup`** **Result:** You receive the following message from `nfs setup`: `Enable Kerberos for NFS?`

2	Enter **`y`** to continue.
	Result: You are asked to specify the type of KDC.
	`The filer supports these types of Kerberos Key Distribution Centers (KDCs):`
	`1 - UNIX KDC`
	`2 - Microsoft Active Directory KDC`
	`Enter the type of your KDC (1-2):`

3	Enter **`2`**.
	Result: You are prompted to specify the filer name.
	`The default name of this filer will be 'SERVER'`
	`Do you want to modify this name? [no]:`

4	Enter **`yes`** to be prompted for a filer name or press **`Enter`** to access the default filer name "SERVER". If you have previously assigned a name to the filer, it will appear in place of the default filer name. **Result:** You are prompted to specify the domain name for the filer's Active Directory server. `Enter the Windows Domain for the filer []:`

Step	Action
5	Enter the domain name for the Active Directory server, as shown below. **Example:** **`ADKDC.LAB.DOCEXAMPLE.COM`** **Result:** The domain name you enter is also used as the Kerberos realm name. You are prompted to set up a local administrator account.
6	Enter the local administrator account information. **Note** This step has no effect on Kerberos configuration for an Active Directory KDC.
7	After you enter local administrator account information, verify that you receive a message that looks similar to the following example. `ADKDC.LAB.DOCEXAMPLE.COM is a Windows 2000(tm) domain.` This message verifies that the filer was able to find the Active Directory server, and that the filer has determined this server can function as a KDC server. If you do not receive a message such as this one, it indicates that there may be a problem with the Active Directory server, or that the DNS server for the filer is not an Active Directory server. Check your network configuration, then run `nfs setup` again.

Step	Action

8	When you receive the following type of message, enter name and password information for the Active Directory domain administrator:
	
	`In order to create this filer's domain account, you must supply the name and password of an administrator account with sufficient privilege to add the filer to the ADKDC.LAB.DOCEXAMPLE.COM domain.`
	`Please enter the Windows 2000 user [Administrator@ADKDC.LAB.DOCEXAMPLE.COM] Password for Administrator:`
	
	Result: If the password is correct and the specified account has the proper permissions within the filer domain, you receive the following type of message:
	
	`CIFS - Logged in as administrator@ADKDC.LAB.DOCEXAMPLE.COM.`
	`Welcome to the ADKDC (ADKDC.LAB.DOCEXAMPLE.COM) Windows 2000(tm) domain.`
	`Kerberos now enabled for NFS.`
	`NFS setup complete.`

Note

You may see the following message in the output text upon completion of NFS setup. This output is an artifact of the installation process, and can be ignored:

```
CIFS is not licensed.
(Use the "license" command to license it.)
```

Setting up NFS Active Directory KDC services after configuring CIFS

To set up Kerberos security services using a Windows Active Directory KDC after CIFS has been set up, complete the following steps.

Note

If you have already used `nfs setup` to enter configuration information, the prompts you receive may differ from those shown in the following procedure.

Step	Action
1	Enter the following command: **nfs setup** **Result:** You receive the following message from `nfs setup`: `Enable Kerberos for NFS?`

2	Enter **y** to continue.
	Result: You are asked to specify the type of KDC.
	`The filer supports these types of Kerberos Key Distribution Centers (KDCs):`
	`1 - UNIX KDC`
	`2 - Microsoft Active Directory KDC`
	`Enter the type of your KDC (1-2):`

3	Enter **2**.
	Result: You receive the following message:
	`Kerberos now enabled for NFS.`
	`NFS setup complete.`
	The Data ONTAP is now configured for Active Directory-based KDC Kerberos over NFS.

Setting up security services using a UNIX-based KDC

To set up Kerberos security services using a UNIX-based KDC, you need to

- Create a principal (a realm user ID) and generate a keytab (key table file) for the filer
- Configure Data ONTAP to use your UNIX-based KDC

Procedures for these tasks are provided in the following sections. These procedures show by example how to add a filer to a standard UNIX-based KDC as a service principal called nfs/*hostname.domain@REALM*.

Note

Due to proprietary restrictions, there are no UNIX-based Kerberos implementations that support CIFS clients. If you configure Data ONTAP for UNIX-based KDC services, be aware that you cannot authenticate CIFS clients.

If you have used previous Data ONTAP versions: In previous Data ONTAP versions, you had to set up Kerberos by using the `options` command to enter values for each Data ONTAP NFS option. Now, you use the `nfs setup` script to set up Kerberos. See Data ONTAP command reference information about NFS Kerberos options (`options nfs.kerberos.*`) you can modify without running `nfs setup`.

Before you set up security services on the filer: 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 for a client and an existing network server.

Note

It is strongly recommended that you enable DNS on your filer before setting up and using secure NFS. If the host component is not already a fully qualified domain name and DNS has not been enabled, then you will need to change all your NFS server principal names in order to enable DNS later.

Creating a principal and generating a keytab file: To create a principal and generate a keytab file, complete the following steps.

Note

The following steps for creating the server principal and keytab on the NFS server are performed using Massachusetts Institute of Technology (MIT) KDC software. If you do not use MIT KDC software, see your software product documentation.

Step	Action
1	If any version of Kerberos is currently enabled on the filer, you must first disable it by running `nfs setup`. On the filer, enter the following command: **`nfs setup`** **Result:** If any form of Kerberos is currently enabled, the following prompt appears: **`Disable Kerberos for NFS?`** Regardless of your response (**`y`** or **`n`**), the filer terminates NFS setup; if you choose to disable Kerberos, the filer first disables any current Kerberos implementation you have configured. For UNIX-based Kerberos, the `nfs.kerberos.file_keytab.enable` option is set to off.
2	On a UNIX or Linux system that supports UNIX-based Kerberos v5 services, enter the `kadmin` command or, if logged into the KDC, enter the `kadmin.local` command.
3	On the `kadmin` or `kadmn.local` command line, 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
4	On the `kadmn` or `kadmn.local` command line, 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 3. *domain* is the domain of the server principal created in Step 3. **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.
5	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 export the /etc subdirectory of the volume. If you export the /etc subdirectory, clients can read the key information and could masquerade as the filer.
6	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`**

Configuring Data ONTAP for UNIX KDC services: Complete the following steps to configure Data ONTAP for a UNIX-based KDC server.

Note

The `nfs setup` script permits you to configure the filer for a UNIX-based KDC before creating the server principal and keytab file. However, you need to create the server principal and keytab file before you can use Kerberos.

Step	Action
1	Enter the following command: **`nfs setup`** **Result:** You receive the following message from `nfs setup`: `Enable Kerberos for NFS?`

2	Enter **`y`** to continue.
	Result: You are asked to specify the type of KDC.
	`The filer supports these types of Kerberos Key Distribution Centers (KDCs):`
	`1 - UNIX KDC`
	`2 - Microsoft Active Directory KDC`
	`Enter the type of your KDC (1-2):`

Step	Action

3	Enter `1`
	
	Result: If you have not yet set up your server principal file and keytab file as described earlier in this procedure, you will receive either or both of the following warnings, but the setup process will continue.
	`There is no /etc/krb5.conf file yet. You will need to establish one.`
	`There is no /etc/krb5.keytab file yet. You will need to establish one.`
	
	If you have previously used a Windows-based version of Kerberos, you may receive a warning to let you know that UNIX-based KDCs cannot validate CIFS clients.
	`It appears that CIFS has been set up to use Kerberos. CIFS requires an Active Directory KDC. If you want to use a UNIX KDC with NFS, you will not be able to secure CIFS with a Kerberos KDC. Do you wish to continue with setup for a UNIX KDC?`
	
	A negative response terminates the `nfs setup` script without saving any of your changes.

4	Enter the Kerberos realm name when you receive the following prompt:
	`Enter the Kerberos realm name.`
	
	The realm name is the realm-specific part of the NFS server's Kerberos principal name (the name you specified for the NFS server principal).
	
	Example:
	
	`NETAPP.COM`
	
	The realm name you enter can be verified or modified later by changing the value of the `nfs.kerberos.realm` option:
	
	`options nfs.kerberos.realm [`*`realm_name`*`]`
	
	Example:
	
	`options nfs.kerberos.realm LAB.NETAPP.COM`

Step	Action

5	Enter a host instance when you receive the following prompt:
	`Enter the host instance of the NFS server principal name [default: server.lab.netapp.com]:`
	Example:
	`server.lab.netapp.com`
	Result: If DNS is enabled, it is used to verify that you have entered a fully qualified domain name for your host. If you have entered a partial name and your host has been entered in DNS, the missing domain information will be appended to your entry.
	The host instance you enter can be verified using the `nfs.kerberos.principal` option.
	`options nfs.kerberos.principal`
	The `nfs setup` script uses your entries for the host instance and realm name to identify the Kerberos principal. The principal is derived from `nfs setup` entries as described below:
	`nfs/<value from nfs.kerberos.principal>@<value from nfs.kerberos.realm>`
	Once you enter the host instance and exit `nfs setup`, the filer is configured to use the key table file you generated. You can modify this configuration later by running `nfs setup` again.

NFS mountd traces

Tracing mountd requests

The `nfs.mountd.trace` command enables you to trace denied mount requests against the filer. Because there is a possibility that the syslog might get hit numerous times during DOS attacks, this option should only be enabled during a debug session.

To enable or disable tracing of denied mount requests, complete the following step.

Step	Action
1	Enter the following command: **`options nfs.mountd.trace on	off`** **Example:** `options nfs.mountd.trace on` **Note** If this option is disabled, then error messages are only logged once per hour.

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. Valid values for *N* are 3 or 4. **Example:** `options nfs.v4.enable on` **Note** This option is turned on by default for 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.
- File delegation—Allows the server to delegate file control to some types of clients for read and write access.
- 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:

- Delegation feature is not supported by every client type.
- 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 volatile file handles.
- Migration and replication are not supported.
- 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.

Note

On new installations, NFSv4 is disabled by default.

How NFSv4 delegation features work on the filer

NFSv4 provides server-to-client delegation for file operations.

Delegation of file operations can reduce latency, particularly in environments that

- Exhibit a high volume of file open and close requests
- Place a heavy load on the server, with a large number of clients and concurrent read sharing operations
- Already strain resources because of a high load or numerous clients
- Already demonstrate file access latency problems
- Have a large number of fast clients trying to access files

Client support for delegation

Delegations are supported only over NFSv4 (not over prior NFS versions). To verify that a specific client type supports delegation, see the client operating system documentation or feature support documentation.

Prerequisites for filer delegation

Delegation of file operations requires the following configuration criteria to be met:

- The requesting client must support delegation and be running NFSv4.
- No other client has the file open for writing or “deny read.”
- There is a recall path present, so the filer can recall the delegation.

Delegation works on files within any style of qtree, whether or not oplocks have been enabled.

Recalling read delegation

Delegation of file operations to a client can be recalled when the lease expires, or when the filer receives the following requests from another client:

- Write to file, open file for writing, or open file for "deny read"
- Change file attributes
- Rename file
- Delete file

Delegation state recovery

When the server reboots, the delegation state is lost. Clients may *reclaim* the delegation state upon reconnection, instead of going through the entire delegation request process again. When a client holding a read delegation reboots, all delegation state information will be flushed from the filer cache upon reconnect. The client must issue a delegation request to establish a new delegation.

Enabling and disabling read delegation

By default, read delegation is disabled. To modify the read delegation setting, complete the following step.

Step	Action
1	Enter the following command: `options nfs.v4.read_delegation on|off [off]`

Note

Delegation options take effect as soon as they are changed. There is no need to reboot or restart NFS.

Enabling and disabling write delegation

By default, write delegation is disabled. To modify the write delegation setting, complete the following step.

Step	Action
1	Enter the following command: `options nfs.v4.write_delegation on|off [off]`

Note

Delegation options take effect as soon as they are changed. There is no need to reboot or restart NFS.

Retrieving delegation statistics

Use the `nfsstat` command to retrieve information about delegation requests, as described in the following sections. Results returned by the `nfsstat` command include delegation requests that have been granted as well as requests that have been denied due to an error. For information about delegation requests your filer has denied, view the system log file.

You can view delegation information:

- Per client
- Per vFiler
- Per filer

Use the procedures described in the following sections to retrieve delegation information.

Viewing delegation statistics for all clients

To view delegation information for all clients, complete the following step.

Step	Action
1	Enter the following command: `nfsstat -h`

The filer will return individual delegation statistics for each client.

Viewing delegation statistics for a specific client

To view delegation information for a specific client, complete the following step.

Step	Action
1	Enter the following command: **nfsstat -h <*hostname* or *ip_address*>**

The filer will return individual delegation statistics for the specified client.

Viewing delegation statistics for a vFiler

To view delegation information specific to a vFiler, complete the following step.

Step	Action
1	Enter the following command: **vfiler run <*filername*> nfsstat -d**

Viewing delegation statistics for a filer

To view delegation information specific to a filer, complete the following step.

Step	Action
1	Enter the following command: **nfsstat -d**

The filer returns the total number of delegations handled by the filer, including current delegations and any that have been recalled. To view only current delegations handled by the filer use the `lock status` command

When a lease expires

When a lease expires the delegation state is revoked and all of the associated state is marked “soft.” That means if the filer receives a conflicting lock request for this same file from another client before the lease has been renewed by the client previously holding the delegation, the conflicting lock is granted. If there is no conflicting lock and the client holding the delegation renews the lease, then the soft locks are changed to hard lock and will not be removed on a conflicting access. However the delegation is not granted again upon a lease renewal.

How the pseudo-fs in NFSv4 affects mount points

NFSv4 uses a pseudo-fs (file system) a 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 interpreted 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 an 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: **`options nfs.v4.id_domain`** ***`domain`***

ACL characteristics using NFSv4

Use the `nfs.v4.acl.enable` option (disabled by default) to control the setting and viewing of NFSv4 ACLs. The `nfs.v4.acl.enable` option does not affect whether an ACL is enforced and does not affect existing ACLs. With NFSv4 enabled, while `nfs.v4.acl.enable` is on, a client using NFSv4 ACLs can set ACLs on files and directories on the system and when a file or subdirectory of a directory with an ACL is created, they inherit appropriate ACL values. While it is off, new files and subdirectories do not inherit the parent ACL values, and the ACL values cannot be viewed from the client.

For access checking, CIFS users are mapped to Unix users and the mapped Unix user and that user's group membership are checked against the ACL.

If a file or directory has an ACL, that ACL is used to control access no matter what protocol—NFSv2, NFSv3, NFSv4, or CIFS—is used to access the file or directory and is used *even if NFSv4 is no longer enabled on the system*.

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. When the shadow source is available, Data ONTAP uses it. The shadow source contains encrypted user information, instead of the password database.

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 user ID (UID), primary group ID (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 55
- "Creating (PC)NFS user entries on the filer" on page 56

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

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 others 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 absolute 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 enable the WebNFS root directory feature:
<tt>options nfs.webnfs.rootdir.set on</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:

- "Configuring CIFS using cifs setup" on page 63
- "File management through Windows administrative tools" on page 67
- "Displaying and changing the description of the filer" on page 68
- "Creating NetBIOS aliases for the filer" on page 69
- "Disabling NetBIOS over TCP" on page 71
- "How to specify a filer's Windows name" on page 72
- "How to change a filer's computer account password" on page 74
- "How CIFS users obtain UNIX credentials" on page 75
- "Sharing directories" on page 99
- "Displaying and changing share properties" on page 107
- "Deleting a share" on page 110
- "CIFS limitations" on page 111
- "Understanding authentication issues" on page 114
- "Using null sessions to access filer storage in non-Kerberos environments" on page 116
- "Understanding local user accounts" on page 119
- "Creating local groups on the filer" on page 121
- "Understanding home directories on the filer" on page 126
- "How Data ONTAP matches a directory with a user" on page 128
- "Setting up user home directories" on page 130
- "Accessing home directories" on page 138
- "Specifying support for wide symbolic links in home directories" on page 141
- "How to stop offering home directories" on page 142
- "How share-level access control lists work" on page 143
- "Specifying how group IDs work with share-level ACLs" on page 144
- "Changing and displaying a share-level ACL" on page 146
- "Changing and displaying file-level ACLs" on page 153

- "Understanding the SID-to-name map cache" on page 156
- "Sending a message to all users on a filer" on page 162
- "Understanding CIFS auditing" on page 164
- "Understanding oplocks" on page 180
- "Managing oplocks" on page 182
- "Displaying CIFS session information" on page 184
- "Timing out idle sessions" on page 186
- "Disabling CIFS service" on page 187
- "Restarting CIFS service" on page 193
- "Reconfiguring CIFS for the filer" on page 194
- "Selecting domain controllers and LDAP servers" on page 198
- "SMB Signing Support" on page 203
- "Tracking Statistics" on page 204

Configuring CIFS using cifs setup

When to use cifs setup

When a valid CIFS license is present, the `cifs setup` command is automatically invoked during the initial setup of your storage appliance. The `cifs setup` command invokes a utility that prompts you for information such as authentication type, lookup services to be used, and so forth.

To learn about using the cifs setup program for initial cifs configuration, including a list of the information you need when running `cifs setup`, see the *Data ONTAP Software Setup Guide*.

In addition to performing initial CIFS configuration, the `cifs setup` command enables you to perform the following tasks:

- Assign or remove WINS servers
- Enter the filer Active Directory site information, if it is not already configured.
- Join the filer to a domain or change domains
- Automatically generate /etc/passwd and /etc/group files when NIS or LDAP is enabled.

Configuring WINS servers

The `cifs setup` utility allows you to make your storage appliance accessible or inaccessible to systems using WINS, by specifying up to four IPv4 WINS servers, or by disabling WINS. However, running `cifs setup` requires that you halt CIFS. A non-disruptive way to modify WINS servers is to enter a comma-separated list of WINS servers using the `cifs.wins_servers` option. Note that this server list is not additive—if you are adding a third WINS server, you must enter all three IP addresses in a comma-separated list, or your existing two WINS servers are replaced by the server you intended to add.

For more information about the `cifs.wins_servers` option, see the *Command Reference Guide*.

Changing the filer's domain

If you have already configured your filer for Windows Domain authentication and you want to move the filer to a different domain, run the `cifs setup` utility.

Note

In order to perform this procedure, you need an administrative account with the same permissions required to add any Windows server to the domain.

To change the filer domain, perform the following steps:

Step	Action
1	If CIFS is currently running, enter the following command: **`cifs terminate`**
2	Run the cifs setup utility: **`cifs setup`**

3	Accept the default values for all settings, until you see the following question:
	`Do you want to delete the existing filer account information? [no] yes`
	Enter “yes” to delete your existing account information.
	Note
	You must delete your existing account information in order to reach the DNS server entry prompt.
	After deleting your account information, you are given the opportunity to rename the filer:
	`The default name of this filer will be 'filer1'.`
	`Do you want to modify this name? [no]:`

Step	Action

4	Press **enter** to keep the current filer name.
	Data ONTAP displays a list of authentication methods:
	Data ONTAP CIFS services support four styles of user authentication. Choose the one from the list below that best suits your situation.
	`(1) Active Directory domain authentication (Active Directory domains only)`
	`(2) Windows NT 4 domain authentication (Windows NT or Active Directory domains)`
	`(3) Windows Workgroup authentication using the filer's local user accounts`
	`(4) /etc/passwd and/or NIS/LDAP authentication`
	`Selection (1-4)? [1]:`

5	Press **enter** for Active Directory domain authentication. You are prompted to enter the Active Directory domain for the filer.
6	Enter the new domain for the filer. Data ONTAP prompts you for the administrator domain account and password.
7	Enter your administrator account and password.
8	Enter the new fully qualified Windows Domain: `Enter the Windows Domain for the filer []:` **`newdomain.netapp.com`**
9	Continue through the remainder of the `cifs setup` questions, accepting the default values until you exit the `cifs setup` utility. Upon exiting, the `cifs setup` utility starts CIFS.

Step	Action

10	Verify your new DNS server configuration:
	`cifs domaininfo`
	Data ONTAP displays domain information, including the new DNS server setting:
	`NetBios Domain:  NEWDOMAIN`
	`Windows 2000 Domain Name: newdomain.netapp.com`

Changing protocol modes

When you have a valid CIFS license and a valid NFS license, the `cifs setup` utility allows you to change your protocol setting, either to NTFS only, or to multiprotocol (mixed) mode (allowing both NFS and NTFS client access). Before you change from NTFS-only to multiprotocol mode, using `cifs setup`, see “Effects of changing an NTFS-only filer to a multiprotocol filer” on page 195, for an alternate, non-disruptive method to change modes. If you do use `cifs setup` to change to multiprotocol mode, files are not immediately available to NFS clients.

To make files available to NFS clients after changing to multiprotocol mode using `cifs setup`, you must also change the root volume qtree security style to `unix`, then use the `chmod` command to permit unix client access as desired.

File management through Windows administrative tools

Scope of Windows administrative tools descriptions

You can accomplish some CIFS file access management tasks from the Data ONTAP command line by using Windows administrative tools, such as Server Manager and User Manager or Active Directory Users and Computers. This chapter describes both command line and Windows NT methods of accomplishing these tasks.

Note

You can also accomplish some tasks described in this chapter using other Windows server management tools, such as the Computer Management snap-in for Microsoft Management Console (MMC).

This chapter assumes that you know how to use Windows network management tools to access your filer.

Using Windows tools with Data ONTAP

The procedures for managing files using Windows tools (such as MMC) are similar to those for managing a Windows server. The procedures in this chapter provide information for Data ONTAP administration tasks that differ from a Windows server.

Case-sensitivity in Data ONTAP

Unlike text you enter through Windows server administration tools, the Data ONTAP command line is case-sensitive. For example, when you specify a volume name in Windows, you can type in either lowercase or uppercase letters. You cannot use Windows tools to create a qtree named Test at the same level as a qtree named TEST, because Windows tools do not make a distinction between these names. You can create and distinguish these two qtrees from the Data ONTAP command line. For information about how such qtrees would appear to Windows clients, see Chapter 7, “Understanding NFS and CIFS file naming,” on page 266.

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.

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. This list allows up to 255 characters, including commas. The /etc/cifs_nbalias.cfg file allows up to 200 entries.

Creating NetBIOS aliases from the command line

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

Step	Action
1	Enter the following command: **`options cifs.netbios_aliases`** ***`name,...`*** **Example:** `options cifs.netbios_aliases alias1,alias2,alias3` **Note** For character limitations of this command, see the preceding section, "Ways to create NetBIOS aliases."
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`

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

In order to disable NetBIOS over TCP, all filer clients must be running Windows 2000 or later. Once you disable NetBIOS over TCP, only Windows 2000 domain controllers and virus scanners can be used.

Disabling NetBIOS over TCP

To disable NetBIOS over TCP, complete the following steps.

Note

Once you disable NetBIOS over TCP, clients no longer receive Data ONTAP notification messages, such as shutdown messages and Vscan warnings.

Step	Action	
1	**If...**	**Then...**
	CIFS is running	Enter the following command: `cifs terminate` You will be prompted with a message allowing you to alert users and specify a delay time before CIFS terminates.
	CIFS is not running	Go to Step 2.
2	Enter the following command: `options cifs.netbios_over_tcp.enable off`	

How to specify a filer's Windows name

Where Windows names are specified

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

- As the argument to the `cifs sessions` command to display information about a Windows user
- In the /etc/usermap.cfg file to map Windows names to UNIX 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 119. 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 are also 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 to change a filer's computer account password

About changing the filer's domain account password

The CIFS option `cifs.weekly_W2K_Password_change` allows you to change the domain account password of a filer belonging to a Windows Active Directory domain once a week, at approximately 1:00 a.m. on Sunday. The CIFS command `cifs changefilerpwd` instructs the filer (either in an Active Directory domain or an NT4 Domain) to change its domain account password immediately.

Changing the filer's domain account password

To change the domain account password of a filer in a Windows Active Directory domain once a week, complete the following step.

Step	Action
1	Enter the following: **`options cifs.weekly_W2K_Password_change enable on|off`**

To make a one-time change of the domain account password for a filer in either a Windows Active Directory or NT4 domain, complete the following step.

Step	Action
1	Enter the following: **`cifs changefilerpwd`** **Result:** The filer responds with the following message. **`password change scheduled.`** The password change is scheduled, and should take place within a minute.

How CIFS users obtain UNIX credentials

About UNIX credentials

When connecting to the filer, a user on a CIFS client receives a CIFS credential. The user must also have one or more UNIX credentials. A UNIX credential consists of a UNIX-style user ID (UID) and group IDs (GIDs). When a CIFS user tries to connect to the filer, Data ONTAP tries to determine 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 184.

How Data ONTAP uses the UNIX credential

Data ONTAP uses the UNIX credential for the following purposes:

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

How Data ONTAP obtains the UNIX credential

Data ONTAP obtains users' UNIX credentials by looking 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. If you accept the default mapping, you do not need to enter this information. 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` option has been specified, the default login name specified for that option is used. See the Data ONTAP command reference for more information about setting the `wafl.default_unix_user` option.

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 primary 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 user names" on page 79.
	All Windows names are the same as UNIX names and you want all CIFS users to be able to 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 92.	
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 90.

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, set the <code>wafl.default_unix_user</code> option to that user, and then go to Step 6.

See “Creating or disabling the default UNIX user account” on page 95.</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 97.
	You do not want unauthenticated users to access the filer	You are done.

How Data ONTAP maps user names

File used for mapping names

Data ONTAP uses the /etc/usermap.cfg file to map user 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 information 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 313.

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 by narrowing the match.

The IP qualifier can be any of the following:

- 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 filer's Windows name" on page 72.

Meaning of the 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 a wildcard character in the domain name: You can use an asterisk (*) as a wildcard character 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 a 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 83.

Use of empty 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 a 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 83.

Use of empty 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

The /etc/usermap.cfg file might include domain names that contain a dot. The following list describes how Data ONTAP interprets these domain names:

- If the filer is installed in a Windows NT domain, the length of the domain name field affects how the domain name is interpreted.
- If the filer is installed in a Windows Active Directory domain, Data ONTAP interprets the domain names in the same way a Windows server would.

Windows NT domain

If the filer is in a Windows NT 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 Active Directory domain

If the filer is in a Windows Active Directory 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 user names

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 permissions. 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.

Verifying NFS clients

For multiprotocol filers, you can restrict NFS access to allow only clients that have been mapped in the usermap.cfg file. This security restriction is probably most appropriate for non-Kerberos environments that primarily serve CIFS clients but want to allow connections from certain known (IP-mapped) NFS clients. See the Data ONTAP command reference description of the `nfs.require_valid_mapped_uid` option for more information.

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 may 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 to map selected accounts 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 95.

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 *Storage Management Guide*.

Creating UNIX accounts for CIFS users with UNIX names

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

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="3">1</td><td>If you use...</td><td>Then...</td></tr>
<tr><td>NIS but not the /etc/passwd file</td><td>Go to Step 2.</td></tr>
<tr><td>The /etc/passwd file but not NIS</td><td>Go to Step 3.</td></tr>
<tr><td>2</td><td colspan="2">Add the UNIX name of each CIFS user to the NIS password map. You are done.</td></tr>
<tr><td>3</td><td colspan="2">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 <code>passwd</code> command to create the /etc/passwd file on the host. Then copy the file from the host to the filer.
Use the SecureShare® Account Migrator program 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 94.</td></tr>
</table>

Migrating CIFS user accounts to the passwd and group files

About SecureShare Account Migrator

SecureShare Account Migrator is a 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 NetAPP on the Web™ (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

You should 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. **Result:** Only users with accounts in the password database can access the filer.	
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 directory.
- 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 directory. For more information about the default UNIX account, see "Creating or disabling the default UNIX user account" on page 95.

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 to connect to the filer using a guest account	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 ""` You are done.	
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.	

Step	Action
4	Create a user account in the NIS password database or the /etc/passwd file to be used by the guest account.

Sharing directories

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 the directories.

Information you need to create a share

When you create a share, you must provide

- The complete path name of an existing folder, qtree, or volume to be shared
- The name of the share entered by users when they connect to the share
- The permission for the share

  You can select from a list of permissions, or enter specific permissions for each user or a group of users.

Optional information for creating a share

When you create a share, you can optionally specify

- A description for the share

  The share description appears in the Comment field when you browse the shares on the network.

If you create the share from the Data ONTAP command line, you can also 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
- Disabling of virus scanning when files in the share are opened
- Disabling of virus scanning when files in the share are opened for read-only access
- Disallowing file caching in the share by Windows clients
- Support for automatic caching of documents and programs in the share by Windows clients
- You can change these attributes at any time after you create a share.

Information you can specify after creating a share

After you have created a share, you can specify these share properties:

- Maximum number of users who can simultaneously access the share

  If you do not specify a number, the number of users is limited 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 111.
- Share-level ACL

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.

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 or UID.

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. Determination of access permissions is based on the UID and primary GID of the NFS user who is trying to access the file.

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 and give write access to users in Engineering and Marketing, you can create a share and give write access to 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, all the users can 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 the Computer Management snap-in for Microsoft Management Console (MMC), 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 Read access. For more information about share-level ACLs, see "How share-level access control lists work" on page 143.

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 checks the share permissions only of the share that has the symbolic link. By default, the filer does share boundary checking for symbolic links to prevent users from accessing files outside the share.

When you create a share from the Data ONTAP command line, you can disable share boundary checking for symbolic links by specifying the option `-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 304.

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.

After you enable wide symbolic links, you need to create Widelink entries in the /etc/symlink.translations file to specify how the filer determines the destination of each wide symbolic link. For more information about how to create Widelink entries and additional requirements, see “About Widelink entries” on page 302. For more information about how to enable support for wide symbolic links in home directory shares, see “Specifying support for wide symbolic links in home directories” on page 141.

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. The umask 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 you can specify a different umask value using `-umask` *`mask`*.

Example: The following example turns off write access for “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 off virus scanning 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*.

About client-side caching

Client-side caching allows files to be cached for offline use on Windows 2000, XP, or 2003 clients. You can specify whether Windows clients are allowed to cache files on a share. You can also specify whether Windows user documents and programs are automatically cached on a share or whether the files must be manually selected for caching. Manual caching is enabled by default for new shares.

When you create a share from the Data ONTAP command line, you can specify the following:

- `-no_caching` to disable client-side caching for the share
- `-auto_document_caching` to enable user documents to be automatically cached on the share
- `-auto_program_caching` to enable programs and user documents to be automatically cached on the share

When you change share properties from the command line, you can specify the following:

- `-no_caching` to disable client-side caching for the share
- `-manual_caching` to enable manual selection of files to be cached on the share
- `-auto_document_caching` to enable user documents to be automatically cached on the share

- `-auto_program_caching` to enable programs and user documents to be automatically cached on the share

Note

You can also set client-side caching properties for a share using the Computer Management application on Windows 2000, XP, and 2003 clients.

Connecting to the filer using MMC

To connect to a filer using the Microsoft Management Console (MMC), complete the following steps.

Step	Action
1	On your Windows server, go to the MMC.
2	Perform the following actions to connect the Computer Management MMC to the filer. **1.** Select Computer Management from the left panel of the MMC. **2.** Click Action (from the menu choices on top of the MMC). **3.** Select Connect to another computer. **4.** Enter the name of the filer on which you created the folder, qtree, or volume in the Another Computer box. Or click browse to browse for the filer. **5.** Click OK. You are now managing the filer using the MMC.

Creating a share from the MMC

To create a share (publish a resource, such as a file or directory) using the MMC, complete the following steps.

Step	Action
1	Create a folder, qtree, or volume on the filer.
2	Connect to the filer using the MMC as described in “Connecting to the filer using MMC” on page 104.

Step	Action
3	Perform the following actions to launch the Share a Folder wizard in the MMC: **1.** Click Computer Management, if it is not selected already. **2.** Double-click System Tools > Shared Folders > Shares > Action > New Share to launch the Share a Folder Wizard. The wording of these menu items may vary slightly, depending on your Windows version.
4	Follow the instructions in the Share a Folder wizard to share the folder, qtree, or volume you created in Step 1. For a list of information you need to successfully complete the Share a Folder Wizard, see “Information you need to create a share” on page 99.

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*] [-no_caching | -auto_document_caching | -auto_program_caching]** 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

Properties you can change

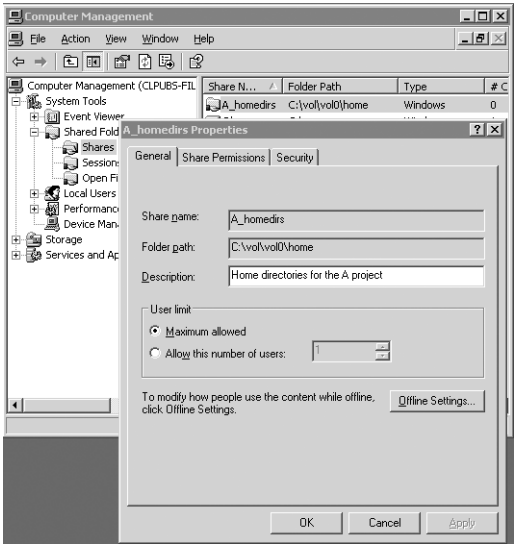
You can change the following share properties:

- Description for the share
- Maximum number of users who can simultaneously access the share
- Share-level permissions

Displaying and changing share properties from the MMC

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

Step	Action
1	Connect to the filer using the MMC as described in “Connecting to the filer using MMC” on page 104.

Step	Action
2	Perform the following actions to display the properties of a share: **1.** Click Computer Management, if it is not selected already. **2.** Double-click System Tools > Shared Folders > Shares. **3.** In the right panel, right-click the share whose properties you want to display. **4.** Select Properties from the drop-down menu. Properties for the share you selected are displayed as shown in the following example. 
3	If you want to change any property, make the appropriate change and click OK.

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:** 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 } { -no_caching | -manual_caching | -auto_document_caching | -auto_program_caching }** Specifying `-nocomment`, `-nomaxusers`, `-noforcegroup`, and `-noumask` removes the description, maximum number of users, forcegroup, and umask value, respectively.

Deleting a share

Deleting a share using the MMC

To delete a share using the MMC, complete the following steps.

Step	Action
1	Connect to the filer using the MMC as described in “Connecting to the filer using MMC” on page 104.
2	Perform the following actions to delete a share: **1.** Click Computer Management, if it is not selected already. **2.** Double-click System Tools > Shared Folders > Shares. **3.** In the right panel, right-click the share that you want to delete. **4.** Select Stop Sharing from the drop-down list. **5.** Click Yes in the Stop Sharing confirmation window. **Result:** The selected share is removed from the list of shares in the right panel.

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 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 filers

The following table shows access limits for the FAS900 series filers.

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 F87 filers

The following table shows access limits for F87 filers.

CIFS limits by filer memory	256 MB 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, R150, and R200 filers

The following table shows access limits for the R100, R150, and R200 filers.

CIFS limits by filer memory	3 GB R100	6 GB R150/R200
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

Understanding authentication issues

Types of authentication

There are three types of authentication for the filer:

- Traditional UNIX authentication
- Windows workgroup authentication
- Windows Kerberos authentication

UNIX Mode authentication

Using UNIX mode, authentication is performed using entries in the /etc/passwd file and/or using NIS/LDAP-based authentication.

Using UNIX authentication:

- Passwords are sent “in the clear” (unencrypted).
- Authenticated users are given credentials with no unique user identification (SID).

The filer verifies the received password against a “hash” (algorithmic variant) of the user password. Passwords are not stored on the filer.

Configuration requirements for UNIX authentication

In order to provide UNIX client authentication, the following items must be configured:

- Client information must be in the filer /etc/passwd file.
- Client information must be entered in NIS and/or ‘LDAP.
- Windows client registries must be modified to allow plain text passwords.

Because UNIX authentication transmits unencrypted passwords, Windows clients require a registry edit to enable them to send passwords without encryption. Clients that are not properly configured to send clear text passwords to the filer may be denied access and display an error message similar to the following:

```
System error 1240 has occurred.
The account is not authorized to login from this station.
```

Enabling plain text passwords for Windows clients

Refer to Microsoft support for information to enable plain text passwords, to allow clients to use UNIX authentication.

Workgroup authentication

Workgroup authentication allows local Windows client access. Workgroup authentication

- Does not rely upon a domain controller
- Limits filer access to 96 local clients
- Is managed using the filer's `useradmin` command

For more information about managing local user accounts in workgroups, see "Understanding local user accounts" on page 119.

Kerberos authentication

Upon connection to the filer, the client will negotiate the highest possible security level. There are two primary levels of security that can be chosen:

- Basic (Windows NT-4) security, based on network services such as NTLM, lanman, and netlogon
- Extended security using Windows 2000 Kerberos implementation

Note

Extended security features are only available to clients that are members of a Windows Active Directory domain.

Using null sessions to access filer storage in non-Kerberos environments

What a null session is

Null session access provides permissions for network resources, such as filer data, to client-based services running under the local system. A null session occurs when a client process uses the “system” account to access a network resource.

Why null session configuration is specific to non-Kerberos authentication

In Kerberos implementations, clients that run local processes using the “system” account assign those processes to the machine account when accessing remote resources. The machine account is assigned the computer name registered with the domain controller, followed by a dollar sign ($). Machine accounts are subjected to the same Kerberos authentication as user accounts, so they do not need to be mapped on the filer.

How the filer provides null session access

Because null session shares do not require authentication, clients that require null session access must have their IP addresses mapped on the filer. By default, unmapped null session clients can access certain filer services, such as share enumeration, but they are restricted from accessing any filer data.

Unless otherwise configured, a client running a local process that requests filer access through a null session is a member of only non-restrictive groups, such as “everyone.” To limit null session access to selected filer resources, you may want to create a group to which all null session clients belong; creating this group enables you to restrict filer access and to set filer resource permissions that apply specifically to null session clients.

How to grant null users access to file system shares

To allow access to filer resources by null session clients, configure your network by performing the following tasks:

- Assign a group to be used by null session clients.
- Record the IP addresses of null session clients to add to the filer’s list of clients allowed to access data using null sessions.

On the filer, you grant null session access to resources by performing the following tasks:

- Configure the /etc/usermap.cfg file to specify the IP addresses of all clients to be allowed access to filer resources using the null sessions.
- Set the `options cifs.mapped_null_user_extra_group` option to the group name you intend to use for null session clients.
- Set permissions to allow appropriate access rights to null session clients.

Data ONTAP provides a mapping syntax in the /etc/usermap.cfg file to specify the IP address of clients allowed access to filer resources using a null user session. Once you create a group for null users, you can specify access restrictions for filer resources and resource permissions that apply only to null sessions.

Caution

Any null user accessing the filer from a mapped IP address is granted mapped user permissions. Consider appropriate precautions to prevent unauthorized access to filers mapped with null users. For maximum protection, place the filer and all clients requiring null user filer access on a separate network, to eliminate the possibility of IP address "spoofing."

Configuring the /etc/usermap.cfg file for null users

To configure the /etc/usermap.cfg file with null users so they have access to null session shares, complete the following steps.

Step	Action
1	Open the /etc/usermap.cfg file.

2	Add an entry for each null user using the following format:
	`IPqual:"" => unixacct`
	`IPqual` specifies either an IP address (hostname or numeric dot-format) or a subnet (IP address + network mask).
	`""` indicates null user.
	`=>` indicates the mapping direction.
	unixacct is the UNIX account (from /etc/passwd or NIS) that the mapped null user will have.
	Examples:
	`10.10.20.19:"" => exchuser`
	`192.168.78.0/255.255.255.0:"" => iisuser`
	Result:
	The client at IP address 10.10.20.19 is allowed null session access to the filer. The null user account is mapped to a UNIX account called *exchuser*, which must exist in the /etc/passwd or NIS database.
	Also, any clients establishing a connection from the 192.168.78.0 class C subnet are allowed null session access and are mapped to the UNIX account *iisuser*. Other null user connections to the filer are not allowed.

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, you should 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 can also use this command to manage non-local users through the domainuser subcommand). 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 *System Administration 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 may use Admin Roles to 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

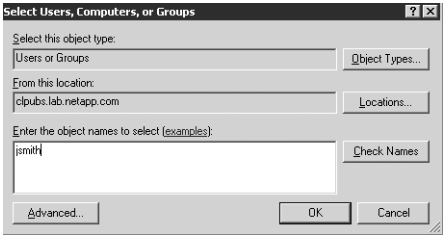
If you use the SnapMirror® feature to copy a volume to another filer and the volume has an ACL for a local group, the ACL does not apply on the mirror. This is because the group is local to the source filer. Because the mirror is a read-only volume and you cannot change ACLs or permissions on it, do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

To create a local group on the filer from the MMC and add users to that group, complete the following steps.

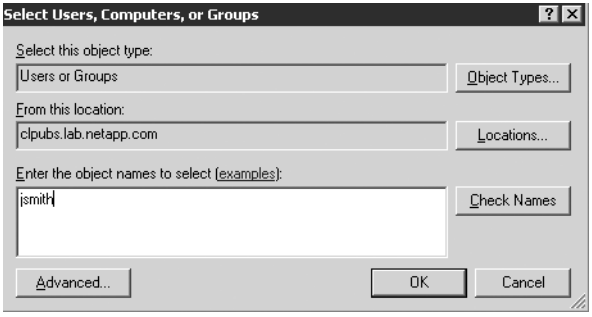
Step	Action
1	Connect to the filer using the MMC as described in “Connecting to the filer using MMC” on page 104.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>2</td><td colspan="2">Perform the following actions to create a local group:
1. Click Computer Management, if it is not selected already.
2. Double-click System Tools > Local Users and Groups.
3. Right-click Groups.
4. Select New Group.
5. Enter the name of the group and a description for it in the New Group window.</td></tr>
<tr><td rowspan="3">3</td><td>If...</td><td>Then...</td></tr>
<tr><td>You do not want to add users to the group</td><td>Click Create.
Result: A new group is created on the filer.</td></tr>
<tr><td>You want to add users to the group</td><td>1. Click Add.
2. In the Select Users, Computers, or Groups window, enter the name of the user in the “Enter the object names to select” box, as shown in the following example.
 

3. Click OK.
4. In the New Group window, click Create.
Result: A new group is created on the filer.</td></tr>
</table>

Adding users to a local group

To add users to a local group on the filer, complete the following steps.

Step	Action
1	Connect to the filer using the MMC as described in “Connecting to the filer using MMC” on page 104.
2	Perform the following actions to add a user to a local group: **1.** Click Computer Management, if it is not selected already. **2.** Double-click System Tools > Local Users and Groups> Groups. **3.** In the right panel, right-click on the group to which you want to add a user. **4.** Select Add to Group. Result: The *groupname* Properties window is displayed. **5.** In the *groupname* Properties window, click Add. **6.** In the Select Users, Computers, or Groups window, enter the name of the user in the “Enter the object names to select” box, as shown in the following example.  **7.** Click OK. **Result:** A user is added to the group you selected on the filer.

Deleting a local group

To delete a local group from the filer, complete the following steps.

Step	Action
1	Connect to the filer using the MMC as described in “Connecting to the filer using MMC” on page 104.
2	Perform the following actions to delete a group: **1.** Click Computer Management, if it is not selected already. **2.** Double-click System Tools > Local Users and Groups> Groups. **3.** In the right panel, right-click on the group that you want to delete. **4.** Select Delete. **5.** Click OK in the Local Users and Groups window. **Result:** The local group is deleted from the filer.

Deleting users from a local group

To delete users from a local group, complete the following steps.

Step	Action
1	Connect to the filer using the MMC as described in “Connecting to the filer using MMC” on page 104.

Step	Action
2	Perform the following actions to delete a user from a group: **1.** Click Computer Management, if it is not selected already. **2.** Double-click System Tools > Local Users and Groups> Groups. **3.** In the right panel, right-click on the group whose user you want to delete. **4.** Select Properties. **Result:** A list of users (members) belonging to that group is displayed in the Members box. **5.** Click the user you want to delete. **6.** Click Remove. **Result:** The selected user is deleted.

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 user a home directory share. The share is called the home directory. From the CIFS client, the home directory works the same way as any other share to which the user can connect.

Each user can connect only to his or her home directories, not home directories 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 128.

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 directory. The home directory for jdoe corresponds to the /vol/vol1/enghome/jdoe directory, and the home directory for jsmith corresponds to the /vol/vol2/mktghome/jsmith directory.

Differences between a home directory and other shares

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

- You cannot change the share-level ACL and the comment for a home directory.
- The `cifs shares` command does not display the home directories.
- The format of specifying the home directory using the Universal Naming Convention (UNC) is sometimes different from that for specifying other shares. For more information about UNC, see “Accessing home directories” on page 138.

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.
- Hidden name—If the naming style is hidden, users connect to their home directories using their Windows user name with a dollar sign appended to it (*name*$), and Data ONTAP searches for a directory that matches the Windows user name (*name*).
- Windows domain name and Windows name—If users from different domains have the same user name, they must be differentiated using the domain name. In this naming style, Data ONTAP searches for a directory in the home directory path that matches the domain name. Then it searches the domain directory for the home directory that matches the user name.

  Example: To create a directory for engineering\jdoe and a directory for marketing\jdoe, you create the two directories in the home directory paths. The directories have the same names as the domain names (engineering and marketing). Then you 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 home directory paths for the directory named johndoe (not jdoe) and offers it as the home directory 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 user's Windows name for directory matching. This is the same style used by versions of Data ONTAP prior to version 6.0.

There are some differences between Windows name directory matching and the directory matching of Data ONTAP versions prior to 6.0. For more information about these differences, see "About symbolic links used as home directory names" on page 129.

About symbolic links used as home directory names

In releases prior to Data ONTAP 6.0, if you wanted home directories to reside in different volumes, you had to specify symbolic links as home directories in the home directory path. Because Data ONTAP now supports home directories in different volumes, you do not need to use symbolic links as home directory names. However, Data ONTAP 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 used them before 6.0. It follows 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. If the path is a symbolic link, 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 297.

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 131
- “Specifying the naming style of home directories” on page 134
- “Creating directories in a home directory path” on page 135

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 for the directory that matches the user name. Data ONTAP stops searching when it finds the matching directory.

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 a subdirectory that is automatically opened when users access their home directories. For an example of adding an extension, see "When Data ONTAP processes changes to cifs_homedir.cfg file" on page 131.

How to specify home directory paths

You can specify home directory paths by editing the /etc/cifs_homedir.cfg file. You can specify up to 1,000 path names in the /etc/cifs_homedir.cfg file.

When Data ONTAP processes changes to 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 by using the `cifs homedir load` command.

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.

Step	Action
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 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. 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` **Note** If you are using the hidden naming style for home directories, when you display the list of home directory paths, Data ONTAP automatically appends a dollar sign to the home directory name (for example, *name*$).

When hidden home directories are not displayed

If you are using the hidden naming style for home directories, home directories are not displayed in the following cases:

- In DOS, when you use the `net view \\filer` command
- In Windows, when you use an Explorer application to access the filer and display home directory folders

Specifying the naming style of home directories

Specifying the naming style of home directories

To specify the naming 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|hidden|domain|mapped|""}`** Use `ntname` if the home directories have the same names as the Windows user names. Use `hidden` if you want to use a Windows user name with a dollar sign ($) appended to it to initiate a search for a home directory with the same name as the Windows user name. 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 for 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 for the users in the same domain. **Example:** If two users have the name 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:** The user with the name engineering\jsmith can attach to the share named jsmith, which corresponds to the /vol/vol0/enghome/engineering/jsmith home directory. The user with the name 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`, `hidden`, `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:** The user with the name engineering\jsmith can attach to the share named jsmith, which corresponds to the /vol/vol0/enghome/engineering/jsmith home directory. The user with the name marketing\jdoe can attach to the share named jdoe, which corresponds to the /vol/vol1/mktghome/marketing/jdoe home directory.

Note

If the naming style is hidden, users must use their user name with a dollar sign appended to it (for example, *name*$) to attach to their 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 /etc/cifs_homedir.cfg file includes 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 directory through Network Neighborhood

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

Syntax for specifying a home directory using a UNC name

Users can also access their home directories using a UNC name. The convention for specifying a home directory when using UNC depends on the home directory naming 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`
`hidden`	*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. When 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:

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

Enabling users to access other users' home directories

Users can only attach their own home directories, not the home directories of other users. To allow users to access all other users' home directories, complete the following steps.

Step	Action
1	Create a share 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 directory using a share alias

For any CIFS home directory naming style, you can connect to your own CIFS home directory 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 home directories

How wide symbolic links in home directories work

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.

Changing support for wide symbolic links in home directory shares

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 131. For details about wide symbolic links, see "About Widelink entries" on page 302 and "Sharing directories" on page 99.

How to stop offering home directories

Disabling home directories

To stop offering home directories, delete the /etc/cifs_homedir.cfg file.

Note

You cannot use the `cifs shares -delete` command to delete home directories.

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.

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

Specifying how group IDs 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 group ID. This is necessary because of the differences between UNIX and Windows security policies involving group permissions.

Example of setting up GID controls for group access

If a share named specs exists in a UNIX-style qtree and you want two UNIX groups, engineering and marketing, to 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 gives this user unrestricted access to the specs share, and access to the draft file is determined by the access rights assigned to the engineering group (`r-x`, in this example).

However, when a member of marketing tries to access the draft file, access is denied because the UNIX-style file permissions grant nonmembers of engineering no access to the file. To make the draft file readable by the marketing group, you need to change the file-level permissions to the following settings:

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

The disadvantage of these permissions is that in addition to marketing, all UNIX users can read the file, 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 to disregard the user's GID when granting access, all users who are not the file's 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 Read access to the special group named Everyone. That is, all users in the domain and all trusted domains have read-only access to the share. You would change the share-level ACL if you want to give users more or less privileges to the share than they are allocated when the share is created.

Methods of changing a share-level ACL

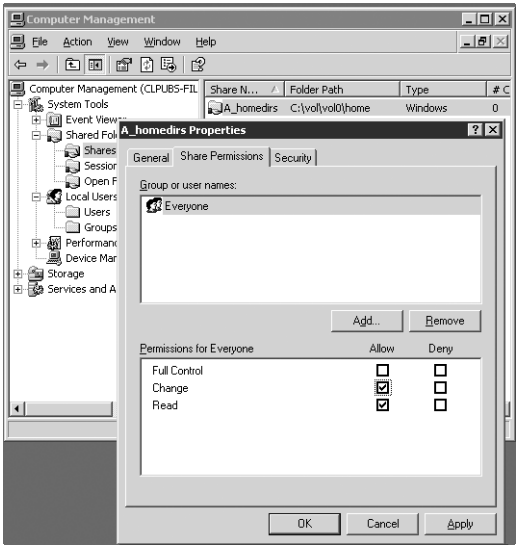
To change a share-level ACL, use one of the following methods:

- From a Windows client, use the MMC. 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.
- 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 at first assumed to be a Windows name, but if the name is not found on the domain controller, the filer tries to look up the name in the UNIX name database.

Displaying and changing an ACL with the MMC

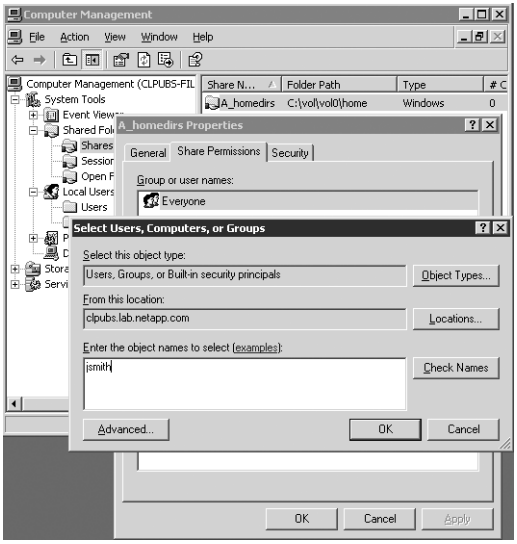
To display and change an ACL with the MMC, complete the following steps.

Step	Action
1	Connect to the filer using the MMC as described in "Connecting to the filer using MMC" on page 104.
2	Perform the following actions to display an ACL of a share: **1.** Click Computer Management, if it is not selected already. **2.** Double-click System Tools > Shared Folders > Shares. **3.** In the right panel, right-click the share whose ACL you want to display. **4.** Select Properties from the drop-down menu. **5.** Click the Share Permissions tab. **Result:** The ACL of the share is displayed.
3	If you want to change the ACL for a user, select the group or user from the "Group or user names" box and change the permissions in the "Permissions for *group* or *user name*" box, as shown in the following example. 

Adding a user or group to an ACL with the MMC

To add a user or group to an ACL with the MMC, complete the following steps.

Step	Action
1	Connect to the filer using the MMC as described in “Connecting to the filer using MMC” on page 104.

Step	Action
2	Perform the following actions to add a user or group to an ACL of a share: **1.** Click Computer Management, if it is not selected already. **2.** Double-click System Tools > Shared Folders > Shares. **3.** In the right panel, right-click the share to whose ACL you want to add the user or the group. **4.** Select Properties from the drop-down menu. **5.** Click the Share Permissions tab. **6.** Click Add. **7.** In the Select Users, Computers, or Groups window, enter the name of the user in the “Enter the object names to select” box, as shown in the following example.  **8.** Click OK. **Result:** A user or a group is added to the ACL.

Removing an entry from an ACL with the MMC

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

Step	Action
1	Connect to the filer using the MMC as described in “Connecting to the filer using MMC” on page 104.
2	Perform the following actions to delete a user or group from an ACL of a share: **1.** Click Computer Management, if it is not selected already. **2.** Double-click System Tools > Shared Folders > Shares. **3.** In the right panel, right-click the share from whose ACL you want to delete the user or the group. **4.** Select Properties from the drop-down menu. **5.** Click the Share Permissions tab. **Result:** A user or a group is deleted from the ACL.

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 may have NTFS-style security.

Files in a FAT file system do not have ACLs; they use UNIX permissions. When viewed from a CIFS client, files without ACLs will not display the Security tab in the file Properties window.

The file system (FAT or NTFS) for a given resource depends upon the filer authentication method and qtree style for that resource.

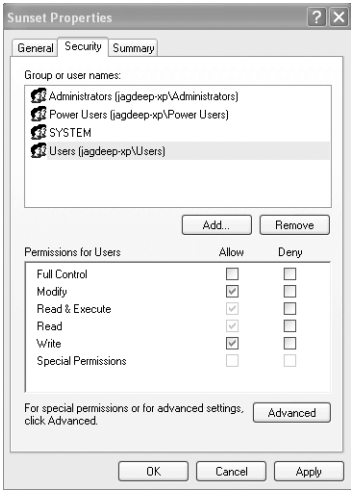
qtree style and authentication method	File system
UNIX-style qtree and all authentication methods	FAT
Mixed or NTFS-style qtree and /etc/passwd authentication	FAT
Mixed or NTFS-style qtree and domain or workgroup authentication	NTFS

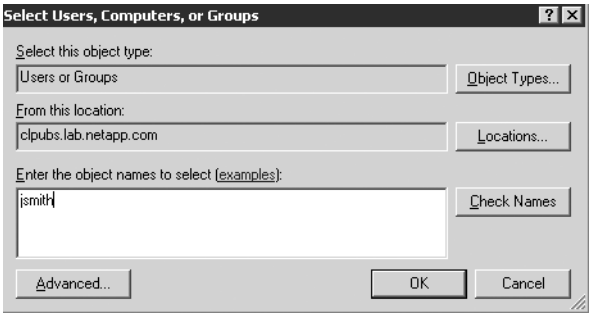
Permission types in a file-level ACL

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

Displaying and setting a file-level ACL

To display and set 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. **Note** Depending on authentication method and qtree style, the Security tab may not be present.
3	Select the user or the group whose permissions you want to display from the “Group or user names” box. **Result:** The permissions for the group or the user you selected are displayed in the Permissions for *user* or *group* box, as shown in the following example. 

Step	Action
4	If you want to add a user or a group to the file, do the following: **1.** Click Add. **2.** In the “Select Users, Computers, or Groups” window, enter the name of the user or the group in the “Enter the object names to select” box as shown in the following example.  **3.** Click OK. **Result:** A user or a group is added to the ACL.

Specifying Group Policy Object security settings on a share

You can use the Windows management tools to specify Group Policy Object (GPO) security settings directly on Data ONTAP shares—however, these security settings are not propagated down the directory hierarchy. That is, when you set a GPO security setting on a share, those settings are not applied to objects within a directory inside that share; files and directories do not “inherit” GPO security settings.

Limitations of Windows NT4 clients on shares that support wide symbolic links

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 158
- "Changing the lifetime of SID-to-name mapping entries" on page 159
- "Clearing the SID-to-name map cache" on page 160

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 users change their accounts or user names.

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 specific 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

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, you can use Server Manager or the Data ONTAP command line to send a message.

Broadcast prerequisites and limitations

Some clients may not receive broadcast messages. The following limitations and prerequisites apply to this feature.

- Windows 95 and Windows for Workgroups clients must have the WinPopup program configured.
- Windows 2003 and Windows XP Service Pack 2 clients must have the messenger service enabled. (By default, it is disabled.)
- Messages to users can only be seen by Windows clients connected using NetBIOS over TCP.

Note

Network configuration can also affect which clients receive broadcast messages.

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.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="4">1</td><td>If you want to...</td><td>Then...</td></tr>
<tr><td>Send a message to all CIFS users connected to the filer</td><td>Go to Step 2.</td></tr>
<tr><td>Send a message to a specific CIFS user connected to the filer</td><td>Go to Step 3.</td></tr>
<tr><td>Send a message to all CIFS users connected to a particular volume</td><td>Go to Step 4.</td></tr>
<tr><td>2</td><td colspan="2">Enter the following command:
<tt>cifs broadcast * "message"</tt></td></tr>
<tr><td>3</td><td colspan="2">Enter the following command:
<tt>cifs broadcast client_name "message"</tt></td></tr>
<tr><td>4</td><td colspan="2">Enter the following command:
<tt>cifs broadcast -v volume "message"</tt></td></tr>
</table>

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. If the specified file does not already exist, Data ONTAP creates the file when it saves information to the file. The directory containing the file, however, must exist; otherwise, an error message 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 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 166
- "Saving and clearing audit events" on page 168
- "Understanding event detail displays" on page 176

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 of the internal log

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.

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

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 preceding table for valid values and units for the size threshold.

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.

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

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 preceding table for valid values and units for the time interval.

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.file.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

Data ONTAP includes SNMP traps to provide a trigger for certain actions (such as notification) based on information about certain 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. Registered clients must have SNMP software that listens 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 NFS access appears instead of an object name, 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™ copy 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.

Write cache data loss considerations when using 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, cached write data may be lost.

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 that owns a file oplock sends a file open request, it is temporarily vulnerable to a "race condition" that can occur if the filer requests an oplock break. To prevent this condition, the filer delays sending an oplock break according to the delay time value (in milliseconds) specified by the `cifs.oplocks.opendelta` option. 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 to that client.

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** Setting the `cifs.oplocks.opendelta` option postpones oplock break requests to clients that have just opened files. 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 30 minutes. You can specify the amount of time that elapses (in seconds) 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 using Windows administration tools 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 188
- "Disabling CIFS for the entire filer" on page 190
- "Specifying which users receive CIFS shutdown messages" on page 192

Disconnecting selected clients

Disconnecting selected clients using MMC

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

Step	Action	
1	Connect to the filer using the MMC as described in "Connecting to the filer using MMC" on page 104.	
2	Perform the following actions to disconnect clients from the filer: **1.** Click Computer Management, if it is not selected already. **2.** Double-click System Tools > Shared Folders> Sessions.	
3	**If...**	**Then...**
	You want to disconnect all clients	**1.** Right-click Sessions. **2.** Select Disconnect All Sessions. **3.** Click Yes in the Shared Folders window.
	You want to disconnect one or multiple clients	In the right panel of the MMC **1.** Right-click the client name. **2.** Select Close Session. **3.** Click Yes in the Shared Folders window.

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 still use with CIFS disabled 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

What reconfiguring CIFS means

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
- The file system used by the filer
- Domain or workgroup to which the filer belongs
- Filer name

Prerequisites for reconfiguring CIFS

The following prerequisites must be true 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. However, this is not necessary if you are using Windows 2000.
- Your storage appliance and the domain controllers in the same domain must be synchronized with the same time source. If the time on the storage appliance and the time on the domain controllers are not synchronized, the following error message is displayed:

  ```
  Clock skew too great
  ```

  For detailed steps on how to set up time synchronization services, see the *Storage Management Guide*.

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 and 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 when you connect 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; any in the /etc/http directory are assigned “Everyone Read”.
- 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.

Step	Action
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 and LDAP servers

About setting domain controllers

Upon startup and as listed below, the filer searches for a Windows domain controller. This section describes how and when the filer finds and selects domain controllers.

The filer searches for domain controllers where any of the following is true:

- The filer has been started or rebooted.
- A CIFS `resetdc` command has been issued.
- Four hours have elapsed since the last search.

Note

Active Directory LDAP servers are searched for in similar manner.

Understanding the domain controller discovery process

The following table describes the domain controller discovery process and priority groups. The filer only progresses to a lower priority group when it has failed to contact all domain controllers in the priority group above it.

Note

For Active Directory environments, site membership is one of the criteria by which the filer selects domain controllers (when no preferred domain controllers are available). Therefore, it is important to have the Sites and Services configured properly (with the filer's subnet information included in the same site as the filer).

Domain controller category	Priority groups: Order in which domain controllers are selected
Preferred Controllers in the `prefdc` list	**Group 1:** Preferred domain controllers are selected by the order in which the controllers appear in the prefdc list.
Favored Controllers that share the same Active Directory site membership with the filer (This category is empty for filers in Windows NT environments.)	**Group 2:** Domain controllers from which a response was received within one second of being pinged, in the order of fastest response time
	Group 3: Domain controllers that did not respond within one second, but share the same subnet as the filer
	Group 4: All non-local domain controllers that did not respond within one second of being pinged
Other Controllers that do not share site membership	**Group 5:** Domain controllers from which a response was received within one second of being pinged, in the order of fastest response time
	Group 6: Domain controllers that did not respond within one second, but share the same subnet as the filer
	Group 7: All non-local domain controllers that did not respond within one second of being pinged

Note

Because site membership is specific to Active Directory domains, there is no "favored" category for Windows NT4 domains, nor for mixed-mode domains in which the filer is configured as an NT4 server. In these environments, all domain controllers found through discovery are assigned the category "other."

Specifying a list of preferred domain controllers and LDAP servers

To specify a list of preferred domain controllers and LDAP servers, 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 or LDAP servers. *address* is the IP address of the domain controller or LDAP server. **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` **Note** To force the filer to use a revised list of preferred domain controllers, or LDAP servers, use the `cifs resetdc` command.

Deleting servers from the prefdc list

After you delete a domain from the prefdc list, you should *always* perform a `cifs resetdc` command to update the filer's available domain controller information, as described in step 2 of the following procedure. The filer does not update the domain controller discovery information from network services when the prefdc list is updated. Failure to reset the domain controller information can cause a connection failure, if the filer tries to establish a connection with an unavailable domain controller (or LDAP server).

Note

The filer does not automatically perform domain controller discovery operations upon restart; restarting the filer does not update the available domain controller and LDAP server list.

To delete preferred domain controllers and LDAP servers, complete the following steps.

Step	Action
1	Enter the following command: **`cifs prefdc delete`** ***`domain`*** *domain* is the domain where the preferred domain controller or LDAP server resides. **Example:** The following example deletes the preferred domain controllers for the lab domain: `cifs prefdc delete lab`
2	Enter the following command: **`cifs resetdc [`*`domain`*`]`** *domain* is the domain you specified in step one. The filer disconnects and searches for a domain controller in the order specified in the revised `prefdc` list.

Troubleshooting domain controller connection

To troubleshoot and observe filer domain to controller traffic, enable the `cifs.trace_dc_connection` filer option. For usage information about this option, see the Data ONTAP command reference.

Displaying a list of preferred domain controllers

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

Note
This procedure also displays LDAP connections.

Step	Action
1	Enter the following command: **`cifs prefdc print [`*`domain`*`]`** *domain* is the domain for which you want to display domain controllers. When a domain is not specified, this 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`

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, or to force domain controller discovery to update the list of available domain controllers, complete the following step.

Note
This procedure also reestablishes LDAP connections, and performs LDAP server discovery.

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. The filer reestablishes a connection using the methodology described in "Understanding the domain controller discovery process" on page 198. **Example:** The following example disconnects the filer from the domain controllers for the lab domain: `cifs resetdc lab`

SMB Signing Support

How SMB signing works in Data ONTAP

Data ONTAP supports SMB signing, when requested by the client. SMB signing helps to ensure that network traffic between the filer and the client has not been compromised. For more information about SMB signing and security issues, see the Microsoft Windows documentation.

Performance impact of SMB signing

Because SMB signing is performed on all communications to and from Windows clients, it can have a significant impact on performance. (SMB signing also occurs on communications to domain controllers that request it, but in this case it is unlikely to have a performance impact.)

Depending on your network, and your filer implementation, the performance impact of SMB signing can vary widely, and can only be verified through testing in your network environment.

Reducing performance impact of SMB signing

If SMB signing is causing performance issues, you can disable SMB signing on some or all of your Windows clients. For information about disabling SMB signing on Windows clients, see the Microsoft Windows documentation.

Tracking Statistics

How to view statistics

Using the `stats` commands, you can view system statistics to track performance. The `stats` command is not specific to CIFS-related statistics. The two `stats` commands that output statistics data are `stats show` (for real-time statistical data and `stats stop` (when you are tracking statistics over a range of time). (Note that the `cifs stats` command is still available.)

The statistics displayed by the `stats` command are accumulated in counters. You reference a specific counter using a hierarchical name with components, *object_name:instance_name:counter_name*, for example, a counter might be named system:system:cifs_ops. You can use the `stats list` command to determine the object_names, instance_names and counter_names available on your NetApp system.

Tracking statistics over a range of time

The output of the `stats` show command provides data describing the filer at the moment you issued the command. To track statistics over time, use the `stats start` command to mark the beginning of the time period you want to track, and the `stats stop` command to mark the end of the time period for which you want to collect statistical data. Data ONTAP outputs the collected data as soon as you enter the `stats stop` command.

Tracking multiple statistics over different time ranges

Data ONTAP allows you to use the `stats start` and `stats stop` command to track different statistics concurrently. To do this, you can enter an instance (`-i`) argument with the `stats start` and `stats stop` commands.

For more information about usage and syntax, see the NetApp stats(1) man page.

To view a list of statistics you can track, complete the following steps:

Step	Action
1	Enter the following command, to view a list of objects that are tracked by the `stats` command: **`stats list objects`** Data ONTAP returns a list of objects you can view using the `stats show` *`object_name`* command.

Step	Action
2	Enter the following command, to view a list of statistics instances: **`stats list instances`** Data ONTAP returns a list of instances you can view using the `stats show` command. You can use these instances to focus the output of the `stats show` command.
3	Enter the following command, to view a list of statistics counters: **`stats list counters`** Data ONTAP returns a list of counters you can view using the `stats show` command.
4	Enter the following command to receive a description of all counters, instances, or objects: **`stats explain counters`** Data ONTAP returns a description of all counters, instances, and objects you can use to focus the output of the `stats show` command.

Once you know the objects, instances, and counters you can monitor to track individual statistics, you can use them as command line arguments to focus the output of the `cifs show` command. For more information, see the NetApp stats(1) man page.

How to view specific statistics

To view statistics for individual objects, instances, and counters, perform the following step:

Step	Action
1	Enter the following command: **`stats show [[<object_name>][:<instance_name>][:<counter_name>]]`** Data ONTAP returns the specific statistics you request.

Saving and reusing statistics queries

You can store and reuse “preset” statistics queries you commonly perform. Preset queries are stored in XML files, in the following location and naming format:

/etc/stats/preset/*preset_name*.xml

For information about how to store and reuse queries, see the NetApp stats_preset(5) man page.

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 208
- "Setting up FTP service on your filer" on page 214
- "Denying access to the FTP service" on page 222
- "Viewing FTP log files and connection statistics" on page 223
- "Number of log files per FTP session" on page 225
- "Managing FTP connections" on page 226

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 215.

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 210.

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 as described in "Anonymous FTP access options" on page 210.

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, authentication goes through the Windows domain controller. This style offers a more secure authentication method than the UNIX style, 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, the domain that the filer is currently in is used.)

The home directory for each user is specified by a combination of the value specified for 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.

In an NTLM environment, FTP users must be authenticated at the highest level of the directory structure in which their home directories are located. For example, if the user's home directory is /vol/vol1/home/user, they must have appropriate permissions for /vol/vol1 and /vol/vol1/home as well as for their home directory.

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 FTP access" on page 221.

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. To enable anonymous FTP access, you must create a user account for Data ONTAP to use for anonymous users. The following table lists the options in Data ONTAP for setting up anonymous access to the filer.

Option	Values (default)	Description
`ftpd.anonymous.enable`	On | Off [Off]	Enables or disables `ftpd` daemon.
`ftpd.anonymous.name`	*name_string* [anonymous]	Enables you to change the default name (*anonymous*) used.
`ftpd.anonymous.home_dir`	*pathname* [] (null)	Specifies the directory to which the anonymous user will be mapped. Anonymous users can access only the directory specified by this option; authenticated users can access all volumes on the filer, unless otherwise configured using the `ftpd.dir.restriction` option. When specifying a pathname for this option, you must first create the directory being specified. (Anonymous home directories are not created automatically.) The default anonymous user account must have read permission for this directory.

Anonymous FTP authentication notes

Anonymous FTP for UNIX authentication: Although the /etc/passwd file contains a special entry, *ftp*, users must use the value specified in 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.

Anonymous FTP for NTLM authentication: The access privileges of a null user apply for anonymous access using NTLM.

Anonymous FTP for mixed authentication: The authentication style used for anonymous access depends upon 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 create a file named ftpusers in the /etc directory and include their user names. 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 222.

Other configurable FTP service options

The following table lists the other configurable options available for the FTP service.

Option	Values (default)	Description
`ftpd.dir.restriction`	On | Off [Off]	When enabled, this option restricts named user access permissions to their home directory or to the override directory, if specified in the `ftpd.dir.override` option.

Option	Values (default)	Description
`ftpd.dir.override`	*pathname* [] (null)	When specifying a pathname for this option, you must create the directory being specified. (Override directories are not created automatically.) The default anonymous user account must have read access permissions for this directory.
`ftpd.idle_timeout`	*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.	Specifies the time in seconds, minutes, or hours after which an idle connection is disconnected.
`ftpd.max_connections`	*number* [500]	Specifies the maximum number of concurrent FTP connections allowed. FTP requests received after this limit has been set will be refused, and the SNMP trap for this value (if it has been set) will be triggered. The upper limit for this value is 5000.
`ftpd.tcp_window_size`	*number* in bytes [32000]	Specifies the TCP window size, which specifies the amount of outstanding unacknowledged data a sender can send on a connection. **Caution** Setting this option inappropriately will adversely impact the FTP performance on your filer. If you are unsure of the value to set, use the default value.

Concurrent FTP connection limitations

The number of available FTP connections depends upon the available filer resources at runtime. Filer services other than FTP may, upon high demand, reduce the maximum number of FTP connections to less than the default value (400) for `ftpd.max_connections`.

Each FTP action (such as `ftpget` and `ftpput`) may use one or two connections, depending upon the operation being performed.

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 215
- "Changing the FTP file size" on page 216
- "FTP file locking" on page 217
- "Configuring an authentication style" on page 218
- "Configuring anonymous FTP access" on page 221

Enabling the FTP service on your filer

About enabling the FTP service

You 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`**

Changing the FTP file size

About changing the FTP file size

You can change the FTP file size on your filer by configuring the `ftpd.log.filesize` option. The default file size is 512K.

Enabling the FTP service

To change the FTP file size on your filer, complete the following step.

Step	Action
1	Enter the following command: **`options ftpd.log filesize [`*`number_and_qualifier`*`]`** **Example:** `options ftpd.log filesize 512k` The file size qualifier (k, K, g, or G) specifies the unit of measurement and is case-insensitive. The minimum is 1K and the maximum is 4G minus one.

When a log file reaches the specified size, Data ONTAP begins writing FTP log information to a new log file. If the number of log files reaches the number specified by the option `ftpd.log.nfiles`, Data ONTAP "wraps around," and begins overwriting information in the first log file.

FTP file locking

About FTP file locking

FTP file locking prevents files from being deleted or renamed while being transferred by FTP from the filer. The FTP file locking option is turned off by default on the filer. When the file is turned off, files being transferred by FTP are not processed by the FTP daemon and could be deleted.

Enabling FTP locking

To enable FTP locking on your filer, complete the following step.

Step	Action
1	Enter the following command: `options ftpd.locking delete`

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 208.

Limitations 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.

NTLMv2 relies on domain controller-based services which do not exist on the filer. For this reason, only NTLMv1 and earlier can be used to connect to filers operating in workgroup mode.

Workgroup filer Windows clients that use NTLM authentication should have "LAN Manager authentication level" set to a level other than "NTLMv2 Only." Setting this option changes the registry value for "LMCompatibilitylevel" to 0, 1, or 2. These are the only NTLM settings supported by the filer for workgroup environments.

Though domain-based clients in an Active Directory environment can authenticate using NTLMv2 (because requests are passed along from the filer to the domain controller), no connection information for local filer accounts will be available to the domain controller. For this reason, local filer accounts would fail authentication during attempts to connect to a filer in such an environment.

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`*** *pathname* is location of user home directories; for example, \home. **Note** The home directory of a user is a combination of the path name 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.

Step	Action
1	Enter the following command: **`options ftpd.auth_style mixed`**

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 208 and “Configuring the UNIX authentication style” on page 219.

Configuring anonymous FTP access

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 210.

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`**

Anonymous FTP sessions are disabled by default. Anonymous FTP sessions are limited to the home directory for anonymous users (and subdirectories inside that directory). The default home directory for anonymous users is specified by the option `ftpd.anonymous.home_dir`, which is set to /vol/vol0/home/ftp by default.

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.

Step	Action
1	Open the /etc/ftpusers file with a text 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 and connection statistics

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.

FTP statistics you can view

Use the `ftp stat` command to view the following FTP statistics:

- Current number of FTP connections
- Highest number of simultaneous FTP connections
- Total number of FTP connections since FTP statistics were reset

Viewing FTP statistics

To view FTP statistics, perform the following step.

Step	Action
1	From the Data ONTAP command line, enter the following command: `ftp stat`

Resetting FTP statistics

To reset FTP statistics, perform the following step.

Step	Action
1	From the Data ONTAP command line, enter the following command: `ftp stat -z` **Result:** All FTP statistics counters are reset.

Number of log files per FTP session

About the log file limit per FTP session

There are two log files per FTP session: ftp.cmd and ftp.xfer. The ftp.cmd is for command log and the ftp.xfer is for the data transfer log. When the log being written reaches the `ftpd.log.filesize` limit, Data ONTAP begins writing to a new log file. Once the number specified by `ftpd.log.nfiles` is reached, Data ONTAP "wraps around" and begins overwriting the first log file.

Changing the number of log files kept in the system

To change the number of log files you want to keep in the system, complete the following step.

Step	Action
1	Enter the following command: **`options ftpd.log nfiles [number of files]`** **Example:** `options ftpd.log nfiles 10` The number you specify must be from 1 to 100, inclusive. **Note** If you change the `ftpd.log.filesize` or the `ftpd.log.nfiles`, you must restart the FTP service using the `options ftpd.enable on|off` command. Because log settings are initiated when the FTP session is started, if you do not restart your FTP sessions, your current log settings are not applied to the filer.

Managing FTP connections

Setting the FTP connection threshold

The FTP connection threshold (`ftpd.max_connections_threshold`) triggers a system log entry when the number of FTP connections approaches the maximum number of allowed FTP connections (`ftpd.max_connections`).

To set the FTP connection threshold, complete the following step.

Step	Action
1	Enter the following command: **`options ftpd.max_connections_threshold n`** *n* is the percentage (0-99) of the value specified for `ftpd.max_connections`. By default, this threshold is set at 0% (off). If the threshold SNMP trap has been set, the trap is triggered each time the value is exceeded.

Available SNMP traps for FTP service

You may want to generate SNMP traps to monitor the following filer information on your client:

- When concurrent connections reach the `ftpd.max_connections_threshold`
- When concurrent connections reach the `ftpd.max_connections`
- When the FTP daemon process stops due to an error

Initializing SNMP traps

To initialize an SNMP trap on the filer and a UNIX client, complete the following steps.

Step	Action
1	At the filer console, enter the following: **`snmp init 1`** **`snmp traphost add [hostname]`**

Step	Action
2	Start `snmptrapd` on the UNIX client that is to receive the trap information. At the client command line, enter the following: **`r snmptrapd -P`**

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 230
- "Testing the HTTP service" on page 231
- "Specifying the filer response to HTTP requests" on page 232
- "Configuring MIME Content-Type values" on page 240
- "Maintaining HTTP service security" on page 242
- "Using virtual hosting" on page 252
- "Displaying HTTP statistics" on page 254
- "Viewing HTTP connection information" on page 258

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.

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`*** You may enter any value from zero (0) to the default value described above.
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 entering 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 filer responses to HTTP requests

You can designate filer responses to HTTP requests to be URL-dependent. 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 233
- "Adding the redirect rule" on page 235
- "Adding the pass rule" on page 237
- "Adding the fail rule" on page 239

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 the component of a URL that you want mapped, 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 and result fields

You can use an asterisk (*) as a wildcard character in the *template* field. The wildcard character matches zero or more characters, including the slash (/) character.

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.

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 and result fields

If you use an asterisk (*) as a wildcard character in the *template* field, the wildcard character matches zero or more characters, including the slash (/) character.

If you use an asterisk as a 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.

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 and result fields

If you use an asterisk (*) as a wildcard character in the *template* field, the wildcard character matches zero or more characters, including the slash (/) character.

If you use asterisks 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.

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 243
- "Using an HTTP virtual firewall" on page 244
- "Protecting Web pages" on page 245
- "Editing the /etc/httpd.access file" on page 247
- "Creating and editing httpd.passwd and httpd.group files" on page 250

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 one or more hosts or on a network interface basis. For more information about these options, see the Data ONTAP command reference information for `protocolaccess`.

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 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 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 domain controller 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:

- You do not need to maintain information in the /etc/http.passwd and /etc/http.group files, thus centralizing user administration.
- 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 transmitted in plain text.

Note

Netscape® browsers send user names and passwords in plain text, providing no security advantage for NTLM.

Editing the /etc/httpd.access file

About the /etc/httpd.access file

The /etc/httpd.access file contains options that govern access to and appearance of each directory. The filer supports the following options:

- Directory—Specifies the directory you want to protect. The directory option encloses all other options.
- 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 options *require user* and *require group* are only required for basic authentication.

Format for option information

Option information for each directory in the /etc/httpd.access file is given in the following format:

```
<Directory directory>
option ...
</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.	
2	Specify the directory tree you want to protect in the following line: **<Directory *directory*>** *directory* is the specific directory tree name you want protected.	
	If...	**Then...**
	You are configuring basic authentication using /etc/httpd.passwd and /etc/httpd.group files	Specify the alias for the directory in the following line: **AuthName *title_phrase*** *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:** `AuthName Secured Area`
	You are configuring NTLM authentication	Specify the following, exactly as shown: **AuthName Windows(tm) Authentication** Go to Step 5.
3	Specify the users who can access the directory in the following line: **require user *user_id* [, *user_id*, ...]** *user_id* is the specific user ID for each user who should have access to the directory.	
4	Specify the groups that can access the directory in the following line: **require group *group_id* [, *group_id*, ...]** *group_id* is the specific group ID for each group that should have access to the directory.	
5	End the option or list of options for the specified directory using the following line: **</Directory>**	

Step	Action
6	Save the file.

Example

The following example shows the use of multiple Directory options 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 and password method to authenticate users, you can 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, you can 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, you can 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, you can 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	In the /etc/httpd.group file, 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 Data ONTAP command reference information for `ifconfig`.

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 of virtual hosting

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 — Greenwich Mean Time (GMT), in `dd/mm/yy:hh:mm:ss` format
- 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 steps.

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/2003:16:45:50] "GET /top.html" 200 1189
192.9.77.2  - - [26/Aug/2003:16:45:50] "GET /header.html" 200 531
192.7.15.6  - - [26/Aug/2003:16:45:51] "GET /logo.gif" 200 1763
198.9.200.2 - - [26/Aug/2003:16:45:57] "GET /task/top.html" 200 334
192.9.20.5  authuser [26/Aug/2003:16:45:57] "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` To revert the setting for log format, change this option from `alt1` to the default value, `common`.

File Access Using WebDAV 6

About this chapter

This chapter describes how you can add WebDAV (Web-based Distributed Authoring and Versioning) protocol to your existing HTTP service.

Note

You can use the WebDAV protocol on your filer as an extension of HTTP *only* if you purchased the license for HTTP. Future versions of Data ONTAP may require the use of a WebDAV license key in order to use WebDAV with HTTP.

Topics in this chapter

This chapter discusses “Understanding the WebDAV protocol” on page 262.

Understanding the WebDAV protocol

WebDAV protocol overview

The WebDAV protocol defines the HTTP extensions that enable distributed Web authoring tools to be broadly interoperable, while supporting user needs. WebDAV allows you to create HTTP directories.

How WebDAV provides support for multiple applications

The WebDAV protocol provides support for remote software development teams though a wide-range of collaborative applications. WebDAV leverages the success of HTTP and acts as a standard access layer for a wide range of storage repositories. HTTP gives read access, WebDAV gives write access.

WebDAV benefits

WebDAV provides a network protocol for creating interoperable, collaborative applications. Major features of this protocol include:

Locking: Long-duration exclusive and shared write locks prevent two or more collaborators from writing to the same resource without first merging changes. To achieve robust Internet-scale collaboration, where network connections may be disconnected arbitrarily, and for scalability, since each open connection consumes server resources, the duration of DAV locks is independent of any individual network connection.

Properties: XML properties provide storage for arbitrary metadata, such as a list of authors on Web resources. These properties can be efficiently set, deleted, and retrieved using the DAV protocol. DASL (DAV Searching and Locating) protocol provides searches of Web resources based on the values in XML properties.

Namespace manipulation: Since resources sometimes need to be copied or moved as the Web evolves, DAV supports copy and move operations. Collections, similar to file system directories, can be created and listed.

HTTP feature support: Data ONTAP WebDAV implementation supports your HTTP configuration settings, such as redirect rules, authentication, and access restrictions. To use WebDAV, you need to have HTTP service enabled and configured. For information about enabling and configuring HTTP, see Chapter 5, “File Access Using HTTP”.

CIFS feature support: Data ONTAP WebDAV implementation supports CIFS home directories when you have valid CIFS and HTTP licenses, and you have enabled WebDAV.

To enable the WebDAV protocol, complete the following step.

Step	Action
1	From a command line, enter the following command: `options webdav.enable on`

HTTP feature support

The Data ONTAP WebDAV implementation supports your HTTP configuration settings, such as redirect rules, authentication, and access restrictions. To use WebDAV, you need to have HTTP service enabled and configured. For information about enabling and configuring HTTP, see “File Access Using HTTP” on page 229.

CIFS feature support

The Data ONTAP WebDAV implementation supports CIFS home directories when you have valid CIFS and HTTP licenses, and you have enabled and configured CIFS home directories. For information about enabling and configuring home directories, see “File Access Using CIFS” on page 61.

Note

The Data ONTAP WebDAV implementation does not support home directory features for virtual IP addresses. URLs that specify a virtual IP address as the host will not be resolved.

To point a WebDAV client to a home directory, complete the following step.

Step	Action
1	In the navigation (or default directory) field of your WebDAV client, enter the URL using the following syntax: **`http://`*`host`*`[:`*`port`*`]/~`** *host* is the host name or IP address for the storage appliance. *port* is the port through which you want to access the storage appliance. ~ is the tilde character (used to specify your user home directory). **Examples:** **`http://eng_filer.lab.company.com/~`** **`http://10.120.83.104:80/~`**

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 266
- "Understanding file locking between protocols" on page 268
- "Understanding read-only bits" on page 269
- "Using LDAP services" on page 271
- "File screening using FPolicy" on page 286
- "Enabling and disabling file screening" on page 288
- "Managing file policies" on page 291
- "Managing file screening servers" on page 296
- "Controlling CIFS access to symbolic links" on page 297
- "Optimizing NFS directory access for CIFS clients" on page 310
- "Preventing CIFS clients from creating uppercase file names" on page 312
- "Accessing CIFS files from NFS clients" on page 313
- "Installing SecureShare Access" on page 325
- "Changing UNIX permissions and DOS attributes from Windows" on page 326

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 following:

- 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 not allowed in MS-DOS file names. Because restrictions on legal characters vary from one operating system to another, see 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. However, if a CIFS user later tries to create spec.txt, the name is not allowed 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, 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 force Data ONTAP to ignore the case in which file names are entered and instead force these names to lowercase text. This option provides better compatibility between 16-bit applications and some UNIX tools. By default, this option is set to ON.

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. For file names that exceed the eight character name or the three character extension limit, 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, one through five, to file names that are no longer unique after being truncated. If it runs out of numbers because there are more than five 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 due to file locking

Because of differences between the NFS and CIFS file 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:

- In mixed or NTFS qtrees, file manipulation operations, such as `rm`, `rmdir`, and `mv`, can cause the NFS application to fail.
- NFS read and write operations are denied by CIFS deny-read and deny-write open modes, respectively.
- NFS write operations 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 any read access file lock set by a CIFS client. 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. However, 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 (disabled) or read-only (enabled).

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 can set 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. Some software, when used by NFS clients and CIFS clients, requires the read-only bit to be enabled.

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:

- NFS treats any file with the read-only bit enabled as if it has no write permission bits enabled.
- If an NFS client disables all write permission bits and at least one of those bits had previously been enabled, Data ONTAP enables the read-only bit for that file.
- If an NFS client enables any write permission bit, Data ONTAP disables the read-only bit for that file.
- If the read-only bit for a file is enabled and an NFS client attempts to discover permissions for the file, the permission bits for the file are not sent to the NFS client; instead, Data ONTAP sends the permission bits to the NFS client with the write permission bits masked.
- If the read-only bit for a file is enabled and a CIFS client disables the read-only bit, Data ONTAP enables the owner's write permission bit for the file.
- Files with the read-only bit enabled are writable only by root.

Note

Changes to a file's permissions take effect immediately on CIFS clients, but might not take effect 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 enabled. Some multiprotocol source control applications require UNIX delete semantics; files for these applications also cannot be deleted when the read-only bit is enabled.

To allow deletion of files using UNIX delete semantics when the read-only bit is enabled, 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.

Using LDAP services

LDAP support in Data ONTAP

Data ONTAP supports LDAP for authentication, file access authorization, and user lookup and mapping services between NFS and CIFS.

This section provides the following information to help you configure Data ONTAP to connect to and query your LDAP database:

- "About using LDAP" on page 272
- "LDAP servers supported" on page 272
- "Specifying the general search base and scope" on page 272
- "Specifying LDAP servers" on page 274
- "Specifying preferred servers" on page 274
- "Enabling or disabling LDAP" on page 275
- "Editing the /etc/nsswitch.conf file for LDAP" on page 275
- "Specifying the administrative user name" on page 275
- "Specifying the administrative password" on page 276
- "Specifying the LDAP port" on page 276
- "LDAP-based UNIX client authentication" on page 277
- "LDAP-based Windows client authentication" on page 277
- "LDAP authorization for NFS file access from Windows clients" on page 277
- "LDAP authorization for NTFS or mixed file system access from UNIX clients" on page 278
- "LDAP server option precedence" on page 278
- "LDAP-based user-mapping services" on page 279
- "Converting to LDAP-based user-mapping" on page 279
- "Configuring Data ONTAP for LDAP-based user-mapping" on page 279
- "Specifying base and scope values for user-mapping" on page 280
- "About Active Directory LDAP servers" on page 281
- "Requirements for Active Directory LDAP servers" on page 281
- "Configuring Data ONTAP for Active Directory LDAP lookup services" on page 281
- "Monitoring Active Directory LDAP server connections" on page 282
- "Active Directory LDAP server connection pooling and selection" on page 283

- "Configuring Data ONTAP for non-Active Directory LDAP servers" on page 284
- "Extending the RFC 2307 schema" on page 284
- "About custom LDAP schemas" on page 284
- "Custom LDAP schema options in Data ONTAP" on page 284
- "Configuring Data ONTAP for custom LDAP schemas" on page 285

About using LDAP

An LDAP server enables you to centrally maintain user information. As a result, you do not have to maintain separate configuration files for each filer that is on your network. If you have several filers on your network, maintaining user information centrally saves you from updating these files on each filer every time you add or delete a user or a group.

If you store your user database on an LDAP server, you can configure your filer to look up user information in the LDAP database.

LDAP servers supported

Data ONTAP LDAP support includes the following types of LDAP servers:

- Netscape®
- iPlanet™
- OpenLDAP
- Windows® Active Directory
- Novell® NDS

Specifying the general search base and scope

The LDAP base is the distinguished name of the LDAP tree in which user information is stored. All lookup requests sent to the LDAP server will be limited to the search base and scope specified by the `ldap.base` option value, unless further restricted by a more specific base and scope lookup value, such as `ldap.base.passwd`, `ldap.base.group`, or `ldap.base.passwd`.

To specify which LDAP base distinguished name to use for looking up user names, complete the following step.

Step	Action
1	Enter the following command: **`options ldap.base name`** *name* is the base distinguished name. Use quotes around names with embedded spaces. **Example:** `options ldap.base "o=networkappliance,c=us"`

Specifying the search base and scope values for user lookups

You can, though it is not required, specify base and scope values specifically for LDAP lookup requests, to limit user lookup queries to the user account branch of your LDAP database. Limiting the search base and scope of every query can significantly improve performance.

To specify base and scope values for all LDAP lookup services, complete the following steps.

Note

The values you assign using this procedure will apply to *all* LDAP lookups, unless you enter separate base and scope values for user-mapping.

Step	Action
1	Set the base and scope search values for password lookups, as they are defined in your LDAP database, by specifying a value for the `ldap.base.passwd` option. **Example:** **`options ldap.base.passwd "ou=People,dc=companydomain,dc=com"`**
2	Set the `ldap.base.group` base and scope search values for password lookups, as they are defined in your LDAP database. **Example:** **`options ldap.base.group "ou=Groups,dc=companydomain,dc=com"`**

Note

Once you specify the search base and scope values for `ldap.base.passwd` and `ldap.base.group`, these values take precedence over the search base and scope set for `ldap.base`, for password and group lookups.

Specifying LDAP servers

To specify the LDAP servers to be used for LDAP queries, complete the following step.

Step	Action
1	Enter the following command: **`options ldap.servers "name[,name...]"`** *name* is the name of an LDAP server. You can enter multiple server names using a space-separated list enclosed in quotes. Data ONTAP attempts to establish connections in the order in which you specify these servers. **Example:** `options ldap.servers "server1,server2"`

Specifying preferred servers

You might want to specify LDAP servers that are on faster links as the preferred servers.

To specify the LDAP servers that you want the filer to attempt to connect to first, complete the following step.

Step	Action
1	Enter the following command: **`options ldap.servers.preferred "name[,name...]"`** *name* is the name of a preferred LDAP server. You can enter multiple server names using a space-separated list enclosed in quotes. **Example:** `options ldap.servers.preferred "server1,server2"`

Enabling or disabling LDAP

To enable or disable LDAP on your filer, complete the following step.

Step	Action
1	Enter the following command: `options ldap.enable {on|off}` Use `on` to enable and `off` to disable LDAP.

Editing the /etc/nsswitch.conf file for LDAP

To edit the /etc/nsswitch.conf file for LDAP, complete the following steps.

Step	Action
1	Open the /etc/nsswitch.conf file on the filer for editing.
2	Enter the following at the password, group, and netgroup lines: `ldap` You can optionally add `files` and/or `nis` to the password line, but they must be entered after `ldap` if you want to use LDAP as the primary mechanism to retrieve user information. **Example:** `passwd: ldap files nis`
3	Save the file.

Specifying the administrative user name

If anonymous authentication does not work in your environment, you need to specify an administrative user name.

To specify the user name to be used for administrative queries for looking up UIDs and GIDs, complete the following step.

Step	Action
1	Enter the following command: **`options ldap.name name`** *name* is the name to be used for administrative queries. The best practice is for this to be the name of a user with read-only access to the LDAP database. Use quotes around names with embedded spaces. **Example:** `options ldap.name "cn=root,o=networkappliance,c=us"`

Specifying the administrative password

To specify a password for the administrative user, complete the following step.

Step	Action
1	Enter the following command: **`options ldap.passwd password`** *password* is the password for the administrative user. The password is displayed as a series of asterisks.

Specifying the LDAP port

You might need to specify the port to use for LDAP queries if the LDAP server has been set up to use a port other than the default for LDAP, port 389.

To specify the LDAP port, complete the following step.

Step	Action
1	Enter the following command: **`options ldap.port N`** *N* is the LDAP port number.

Note

Data ONTAP does not support using LDAP with the Secure Socket Layer (SSL) or Transport Layer Security (TLS) protocol, which is commonly called Secure LDAP or LDAPS. Configuring Data ONTAP to connect to an LDAPS server on port 636 does not provide secure LDAP access for filer queries and, in most cases, will cause all LDAP queries from the filer to fail.

LDAP-based UNIX client authentication

To enable authentication of UNIX clients through an LDAP server, make sure that LDAP is the first protocol entered on the password line of the /etc/nsswitch.conf file as described in "Editing the /etc/nsswitch.conf file for LDAP" on page 275.

LDAP-based Windows client authentication

You can authenticate Windows clients through an LDAP server. To enable authentication of Windows clients through an LDAP server, complete the following additional operations.

- Run `cifs setup` on the filer to be accessed, and specify NIS/LDAP as the authentication method to be used for CIFS clients on that filer.
- Configure the local security settings of each Windows client to use clear text (unencrypted) password authentication rather than use Kerberos or other encrypted authentication methods.
- Verify that your Windows clients have their `userpassword` attribute configured in the LDAP user database.

LDAP authorization for NFS file access from Windows clients

To enable authorization of Windows client access to UNIX files on a filer that uses LDAP authentication, complete the following tasks:

- On the filer to be accessed, verify that the /etc/nsswitch.conf file specifies `ldap` as one of the passwd entries. See "Editing the /etc/nsswitch.conf file for LDAP" on page 275.
- On the filer to be accessed, verify that every CIFS user who needs to access UNIX files is mapped to an associated UNIX user name in the usermap.cfg file.
- Verify that every associated UNIX user name has an entry in the LDAP database.

LDAP authorization for NTFS or mixed file system access from UNIX clients

To support authorization of UNIX client access to an NTFS or mixed file system on a filer that uses LDAP authentication, complete the following tasks:

- On the filer to be accessed, verify that the /etc/nsswitch.conf file specifies `ldap` as one of the passwd entries. See "Editing the /etc/nsswitch.conf file for LDAP" on page 275.
- Verify that every UNIX user that needs to access an NTFS or mixed file system has an entry in the LDAP database.
- On the filer to be accessed, verify that every UNIX user that needs to access an NTFS or mixed file system is mapped to an associated CIFS user name in the usermap.cfg file.

LDAP server option precedence

Data ONTAP chooses an LDAP server based on your LDAP server option settings, as described in the following table.

Server designation option	Server selection order
`ldap.preferred.servers`	When specified, servers listed in this option value will be tried first, according to list order.
`ldap.servers`	When no `ldap.preferred.servers` are specified, or specified servers are not available, servers designated in this option value will be tried, according to list order.
`ldap.ADdomain`	When no `ldap.preferred.servers` and no `ldap.servers` are specified or available, servers designated in this option value will be tried using domain controller selection methodology. For information about domain controller selection methodology, see "Selecting domain controllers and LDAP servers" on page 198.

LDAP-based user-mapping services

You can use LDAP services to map between UNIX and Windows user accounts, instead of using NIS data or to adding entries to the usermap.cfg file. By default, Data ONTAP uses the same (one-to-one) user account resolution process in both directions: UNIX-to-Windows mapping and Windows-to-UNIX mapping.

Converting to LDAP-based user-mapping

When converting to LDAP from file-based user-mapping, you must remove mapping entries (except for null session entries) from the usermap.cfg file. If mapping entries are present in that file, they will be used for user-mapping instead of LDAP records.

Note

If you've configured Data ONTAP for null sessions, make sure you leave the null session client entry in the usermap.cfg file. For more information about null session configuration, see "Using null sessions to access filer storage in non-Kerberos environments" on page 116.

Configuring Data ONTAP for LDAP-based user-mapping

By default, LDAP-based user-mapping is disabled. (Data ONTAP retrieves user-mapping information from the etc/usermap.cfg file.)

Note

To allow Data ONTAP access to LDAP lookup services, if your UNIX user account information is stored in a non-Active Directory LDAP server, that LDAP server must be configured to allow either simple authentication or anonymous user searches.

To enable LDAP-based user-mapping, complete the following steps.

Step	Action
1	From the Data ONTAP command line, specify a value for the option `ldap.usermap.attribute.windowsaccount`. **`options ldap.usermap.attribute.windowsaccount account_name`** *account_name* is the user object attribute Data ONTAP will use for Windows account lookups.
2	Extend your LDAP schema to include the user object attribute you entered in Step 1.

Step	Action
3	From the Data ONTAP command line, specify a value for the option `ldap.usermap.attribute.unixaccount`. **`options ldap.usermap.attribute.unixaccount`** ***`account_name`*** *account_name* is the user object attribute Data ONTAP will use for UNIX account lookups.
4	Extend your LDAP schema to include the values you entered in Step 2 and Step 3.
5	Enter the following command: **`options ldap.usermap.enable on`** If you have a significant load on your LDAP server, you may want to improve performance by setting a separate search base or search base and scope for user-mapping, as described in the following section.

Specifying base and scope values for user-mapping

LDAP options allow you to set search base and scope, to limit attribute searches to the appropriate areas of your LDAP database. Setting these options will improve the speed of LDAP lookups.

Use the following syntax when specifying search base and scope. Base and scope values must correspond to the structure of your LDAP data.

`options ldap.usermap.base "`*`base`*`[:`*`scope`*`][;`*`base2`*`[:`*`scope2`*`]]"`

Example 1:

options ldap.usermap.base ou=People,dc=domain0”

Entering this command sets the search base for user-mapping lookups to `ou=People,dc=domain0` and the (unspecified) search scope defaults to SUBTREE.

Example 2:

`options ldap.usermap.base "(ou=People,dc=domain0):BASE;o=org"`

The use of parentheses applies the specified search scope (`BASE`) to `ou=People,dc=domain0`. The unspecified search scope for the `o` (“org”) object defaults to SUBTREE.

For more information about setting search base and scope values, see your LDAP documentation.

About Active Directory LDAP servers

Data ONTAP provides the ability to connect to Active Directory for LDAP lookup services. To use Active Directory for LDAP services, enter the fully qualified Active Directory domain into the Data ONTAP `ldap.ADdomain` option.

As Windows-to-UNIX mapping is performed using Active Directory, Data ONTAP does the following:

- Verifies that the user account exists within the Active Directory domain specified for that account
- Performs a query to the Active Directory domain specified in the `ldap.ADdomain` option
- Returns the UNIX user account information and verifies that the user account exists

Requirements for Active Directory LDAP servers

In order to use Active Directory as your LDAP server, you must

- Have a valid CIFS license
- Have your filer joined to an Active Directory domain
- Have a two-way trust relationship established between your filer's domain and your LDAP server's domain, if they are different

Configuring Data ONTAP for Active Directory LDAP lookup services

To specify Active Directory LDAP lookup services, complete the following steps:

Step	Action
1	If your UNIX user account information is in Active Directory, or if it is in a non-Active Directory LDAP server that is configured to allow anonymous user searches, go on to Step 2. To configure Data ONTAP for simple LDAP server authentication, enter the user name and password to be used for LDAP lookups into the `ldap.name` and `ldap.passwd` options, respectively. **`options ldap.name user_name`** **`options ldap.passwd password`**

Step	Action
2	In the etc/nsswitch.conf file, specify **ldap** for the passwd entry, the group entry, or both, to designate LDAP as the lookup service to use.
3	If you have a custom schema, enter values for NSSMAP options, as described in the section "Configuring Data ONTAP for custom LDAP schemas" on page 285.
4	From the Data ONTAP command line, enter the following command: **options ldap.ADdomain *fully_qualified_domain_name*** **Note** The domain you enter must either be the local domain or a domain that shares a trust relationship with the local domain. **Example:** **options ldap.ADdomain group.company.com**

Monitoring Active Directory LDAP server connections

Displaying connection status for all LDAP server types: To display information about LDAP server connection status for all server types, complete the following step.

Step	Action
1	From the Data ONTAP command line, enter the following command: **netstat** **Result:** Both Active Directory and non-Active Directory LDAP server connection state information is shown on port 389 (or the non-default value assigned using the ldap.port option).

Displaying Active Directory LDAP server information: To display information about current Active Directory LDAP server connections and Active Directory LDAP server selection preferences, complete the following step.

Step	Action
1	From the Data ONTAP command line, enter the following command: `cifs domaininfo` **Result:** Following the list of domain controller connections and domain controller selection preferences, a list of Active Directory LDAP server connections is displayed, followed by the list of LDAP server selection preferences.

Troubleshooting Active Directory LDAP server connections: To log Active Directory LDAP server connection attempts, complete the following step.

Step	Action
1	From the Data ONTAP command line, enter the following command: `options cifs.trace_dc_connection` **Result:** Information about attempts to connect Active Directory LDAP servers are entered in the system log.

Active Directory LDAP server connection pooling and selection

Data ONTAP performs the following operations, to improve LDAP performance:

- Active Directory LDAP server connections are pooled on a per-domain basis.
- When no response is received from the current LDAP server, subsequent connections are made to the next best available LDAP server.
- Once every minute, Data ONTAP performs a check to see whether a better LDAP server has become available.
- Every four hours, Data ONTAP discovers the available Active Directory LDAP servers and reorders the list, sorting servers in the following order:
  - Preferred servers, left in the order specified by the `prefdc` command.
  - Favored servers, sorted by fastest response time
  - Other Active Directory LDAP servers, sorted by fastest response time

Configuring Data ONTAP for non-Active Directory LDAP servers

Data ONTAP provides the ability to designate one or more LDAP servers by entering one or more IP addresses or fully qualified domain names, separated by commas, as the values for these two Data ONTAP options:

- `ldap.preferred.servers`
- `ldap.servers`

Data ONTAP connects to servers specified by these two option values and attempts to authenticate using a simple bind. Because simple binds do not provide sufficient authentication to establish a connection with Active Directory servers, do not specify Active Directory servers within these two option values.

Extending the RFC 2307 schema

Your RFC 2307-compliant schema must be extended on the LDAP servers that you want to use for LDAP queries.

For more information refer to RFC 2307 or to documentation by third party directory integration vendors.

About custom LDAP schemas

By default, Data ONTAP supports LDAP servers that comply with RFC 2307, which specifies a Network Information Service (NIS)-style schema. You can replace the default values of LDAP options with your custom attribute names to configure Data ONTAP to query your custom (not RFC 2307-compliant) schema.

Custom LDAP schema options in Data ONTAP

The following new options are set by default to the attribute names specified in RFC 2307.

Option	Default value (per RFC 2307)
`ldap.nssmap.objectClass.posixAccount`	`posixAccount`
`ldap.nssmap.objectClass.posixGroup`	`posixGroup`
`ldap.nssmap.attribute.groupname`	`cn`
`ldap.nssmap.attribute.netgroupname`	`cn`
`ldap.nssmap.attribute.nisNetGroupTriple`	`nisNetGroupTriple`
`ldap.nssmap.attribute.memberUid`	`memberUid`
`ldap.nssmap.attribute.uid`	`uid`

Option	Default value (per RFC 2307)
ldap.nssmap.attribute.uidNumber	uidNumber
ldap.nssmap.attribute.gidNumber	gidNumber
ldap.nssmap.attribute.userPassword	userPassword
ldap.nssmap.attribute.homeDirectory	homeDirectory
ldap.nssmap.attribute.loginShell	loginShell
ldap.nssmap.attribute.gecos	gecos

Configuring Data ONTAP for custom LDAP schemas

From the command line, you can modify the string value of Data ONTAP custom schema options to match the corresponding objects in your LDAP schema.

For example, for a custom LDAP schema in which the object containing Group ID (GID) numbers is “groupid,” you would enter the following command:

```
options ldap.nssmap.attribute.gidNumber groupid
```

To configure Data ONTAP to query your custom schema, complete the following step.

Step	Action
1	Configure the Data ONTAP `ldap.nssmap` options to match your custom LDAP server schema. **Example:** `options ldap.nssmap.objectClass.posixAccount Users` Specify values for each Data ONTAP schema option that corresponds to a customized object in your LDAP schema.

File screening using FPolicy

Understanding FPolicy

The FPolicy™ feature allows you to create file policies that specify file operation permissions according to file type. For example, you can restrict certain file types, such as .jpg and .mpg files, from being stored on the filer.

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

What a file policy is

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

Prerequisites for FPolicy

File screening server: FPolicy implementation requires a server with file screening software that is supported by Data ONTAP. You can check for client file screening software on the NetApp Partners Web page at http://www.netapp.com/partners.

Licensing: FPolicy requires CIFS to be licensed and running, even in NFS-exclusive environments. To apply file policies to NFS files, you must also have NFS licensed and running.

Limitations of FPolicy

The following limitations apply to FPolicy:

- Policies are applied to NFS and CIFS files only; policies will not be applied to files created by clients using other protocols.
- You can create and use up to 20 file screening policies at one time.
- Names for screening policies and policy types can have up to 80 characters.

How file screening works

You use file screening policies to specify files or directories with restrictions to be placed on them. Upon receiving a file operation request (such as open, write, create, or rename), Data ONTAP checks its file screening policies before permitting the operation. 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.

FPolicy with vscan

FPolicy runs independently from vscan operations. File policy occurs before virus scanning operations, so that paths indicated in “stub” files (such as symbolic links) can be traversed to load the actual file, instead of just scanning the stub file. Vscan operations are independent of file policies. That is, vscan can open and scan files that have been blocked by file policies.

Enabling and disabling file screening

Enabling and disabling FPolicy

All FPolicy 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]`**

Note

You must have a supported third-party file screening server running and accessible in order to use FPolicy.

Creating a file policy

To create a file policy, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy create`** ***`PolicyName PolicyType`*** *PolicyName* is the policy name for the file screening policy you want to create. `PolicyType` is the policy group to which this file screening policy will belong. Currently, the only supported policy type is “screen”. **Example:** **`fpolicy create policy1 screen`** **Note** To use the policy you’ve created, make sure to enable the policy and to enable file screening using the `fpolicy.enable` option. Also, verify that a file screening server is available.

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`** *PolicyName* is the policy name for the file screening policy you want to enable. **Example:** **`fpolicy enable policy1`** **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`** *PolicyName* is the name of the file screening policy you wish to disable. **Example:** **`fpolicy disable policy1`**

Deleting a file policy

To delete a file policy, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy destroy PolicyName`** *PolicyName* is the policy name for the file screening policy you want to delete. **Example:** **`fpolicy destroy policy1`**

Managing file policies

Displaying all policies and FPolicy status

To display a list of all policies and FPolicy status, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy`**

Displaying information for a file policy

To display information about a specific file policy, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy show`** ***`PolicyName`***

Example of file screening policy information display

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

```
fpolicy show FPOLICY1

File policy FPOLICY1 (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 to screen:
JPG, MPG
List of extensions not to screen:
TXT, LOG

Number of files screened: 1032
Number of failures: 0
```

How the file policy specifies which files to screen

The file policy specifies which files to screen using a list of file extensions to include for screening or to exclude from screening. From the command line, you can display or change the list of included and excluded file extensions. When there is a contradiction for a specific file extension, either by two different policies or by addition to both the include and exclude list on the same policy, that file type will be screened.

You can use a question mark (?) to specify a wildcard character within any file extension you enter for file policy screening commands. For example, entering **`.jp?`** in a list of file extensions to include for file screening would include all file extensions that begin with ".jp" (such as .jpg and .jpe extensions).

Displaying the list of included or excluded file extensions for a policy

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

Step	Action
1	Enter the following command: **`fpolicy ext[ensions] exc[lude] show`** ***`PolicyName`*** When you enter this command, Data ONTAP responds with a list of extensions from the exclude list for the file you specified. **Note** If you want to show file extensions from the list of files to be *included* for file screening, use the include (`inc`) option in place of the exclude (`exc`) option during this procedure.

Including file extensions for file screening

To add file extensions to the list of file extensions to be screened for a file policy, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy ext[ensions] inc[lude] add PolicyName ext[,ext]...`** The file extensions you add to an include list will always be screened by the file screening server when that policy is enabled. **Example:** **`fpolicy ext inc add imagescreen jpg,gif,bmp`** **Result:** Once enabled, the policy `imagescreen` now requires screening for any files with the extension `.jpg`, `.gif`, or `.bmp`.

Excluding file extensions from file screening

To add file extensions to the list of file extensions to be excluded from file screening for a file policy, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy ext[ensions] exc[lude] add PolicyName ext[,ext]...`** The file extensions you add to an exclude list will not be screened by the file screening server when that policy is enabled (unless contradicted by another enabled file screening policy). **Example:** **`fpolicy ext exc add default txt,log,hlp`** **Result:** When enabled, the modified policy will no longer require `.txt`, `.log`, and `.hlp` files to be screened by the file screening server.

Removing specific extensions from a file policy include or exclude extension list

To remove file extensions from the *exclude* or *include* extensions list for a file screening policy, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy ext[ensions] exc[lude] remove PolicyName ext[,ext]...`** **Example:** **`fpolicy ext exc remove default wav`** **Result:** Files with a .wav extension are screened. **Note** If you want to delete specific file extensions from the list of files to be *included* for file screening, use the include (`inc`) option in place of the exclude (`exc`) option during this procedure.

Removing all entries for a file policy include or exclude extension list

To delete all entries from the *exclude* or *include* list for a file policy, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy ext[ensions] exc[lude] reset PolicyName`** **Result:** The list of excluded file extensions is reset to the default (an empty list), so no file extensions are exempted by this policy; therefore, all files are screened when this policy is the only policy enabled. **Note** If you want to delete all file extensions from the list of files to be *included* for file screening, use the include (`inc`) option in place of the exclude (`exc`) option during this procedure.

Replacing the list of excluded file extensions

To replace the entire *exclude* or *include* list for a file policy, complete the following step.

Step	Action
1	Enter the following command: **fpolicy ext[ensions] exc[lude] set *PolicyName* *ext*[,*ext*...]** **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 want to replace the list of extensions to be *included* for file screening, use the include (`inc`) option in place of the exclude (`exc`) option during this procedure.

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 *PolicyName* required on** **Note** If you turn on required file screening for a policy when no file screening servers are available, access to files specified in that policy will be denied. This command is set to `off` by default.

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`** When you enter this command, Data ONTAP returns the status of the file screening server(s) for the policy you specified.

Designating secondary file screening servers

To designate a list of secondary servers to be used when the primary file screening server is unavailable, complete the following step.

Step	Action
1	Enter the following command: **`fpolicy options PolicyName secondary_servers IPaddr[,IPaddr,...]`** When you enter this command, Data ONTAP will choose the fastest responding secondary server whenever the primary screening server is unavailable.

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`** *PolicyName* is the name of the file screening policy *IPaddr* is the IP address of the file screening server.

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 298
- “About Map entries” on page 301
- “About Widelink entries” on page 302
- “About disabling share boundary checking for symbolic links” on page 304
- “Redirecting absolute symbolic links” on page 306
- “Preventing access by CIFS clients to cyclic directory structures” on page 309

Understanding how CIFS clients interact with symbolic links

Enabling CIFS clients to follow symbolic links

The `cifs.symlinks.enable` option, enabled by default, permits CIFS clients to follow symbolic links.

To enable CIFS access to symbolic links after they have been disabled, complete the following step:

Step	Action
1	Enter the following command to enable CIFS access to symbolic links: `options cifs.symlinks.enable on` **Result:** CIFS clients will directly follow relative symbolic links to destinations in the same share.

There are special requirements to enable CIFS access to the following types of symbolic links:

- Absolute symbolic links

  Since the destination of an absolute symbolic link depends on the type of UNIX mount, CIFS clients need additional information to interpret absolute symbolic links.
- Relative symbolic links to destinations on the same filer outside the share in which the relative symbolic link is located

  By default, Data ONTAP does 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 doing one or more of the following:

- 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 301
- “About Widelink entries” on page 302
- “About disabling share boundary checking for symbolic links” on page 304

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.

Note

CIFS client users who follow symlinks to resources outside the link's share do not work, unless the `cifs share -nosymlink_strict_security` option has been specified for the source 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 symbolic link destination must be in the same share as the link itself, or the link must be in a share for which the `-nosymlink_strict_security` option has been specified.

Using Map entries

When you use Map entries to redirect absolute symbolic links, Windows 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 306.

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.

How CIFS clients follow 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:

- Even if the destination of the wide symbolic link is a file, it appears 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, you should 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 306. For information about how to enable a share for wide symbolic links, see “Sharing directories” on page 99.

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 filer. This behavior is set on a per-share basis and affects both relative and absolute symbolic links.

Requirements for disabling share boundary checking

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 of disabling share boundary checking

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 a user can create 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 page 99.

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 301. For information about Widelink entries, see "About Widelink entries" on page 302.

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 cyclic directory structures

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 created in non-Unicode format and directories created by CIFS clients are in Unicode format. Because of this, CIFS directories are directly accessible to NFS clients, but NFS directories are not directly accessible to CIFS clients. To provide a CIFS client with access to an NFS directory, the filer must first convert the NFS directory to Unicode format. This is done automatically ("on the fly"), as the filer receives the access request. Depending on the amount of data involved, Unicode conversion can take time and consume filer resources.

Optimizing CIFS client access to an NFS directory

You can optimize CIFS client access to an NFS directory by performing the following tasks to reduce or eliminate latency caused by converting an NFS directory format conversion:

- Configure Data ONTAP to convert non-Unicode directories to Unicode format when either CIFS clients or NFS clients access directories.
- Change Data ONTAP to create only Unicode-formatted directories, thereby eliminating the need to convert formats.

Note

If you intend to share files between CIFS and NFS clients, configure Data ONTAP to create directories in Unicode format immediately after installing Data ONTAP. This will to ensure that all new directories are created in Unicode format.

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: `vol options volume_name create_ucode on`

Converting to Unicode format

By default, Data ONTAP performs Unicode conversion of a directory only when a CIFS client requests access. You can reduce the time required for Unicode conversion by limiting the number of entries in each directory to less than 50,000.

Once you already have large directories, you can minimize the performance impact of Unicode conversion by preemptively forcing Unicode conversion for large directories as described in the procedure below.

To force Unicode conversion of large directories and to convert directories to Unicode format when they are accessed from both CIFS *and* NFS clients, complete the following steps.

Step	Action	
1	**If...**	**Then...**
	You have a directory that contains more than 50,000 files	**1.** Create a new CIFS directory from a Windows client on the same volume and in the same qtree as the directory you want to convert. **2.** Use the NFS `mv` command to move the files into the directory you just created. **3.** Optionally remove the old directory and assign its name to the new directory. Go on to Step 2, below.
	You have directories that contain fewer than 50,000 files	Go to Step 2.
2	Enter the following command: **`vol options volume_name convert_ucode on`** **Result:** Unicode conversion is performed when NFS clients access files.	

Note

Do not enable the `convert_ucode` option when you have directories that contain more than 50,000 files.

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 user name, and one or more group SIDs that correspond to Windows 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 314
- "Deleting mapping entries from the WAFL credential cache" on page 315
- "Setting how long mapping entries are valid" on page 317
- "Monitoring WAFL credential cache statistics" on page 318
- "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

The cache is limited to 10,000 entries. If you exceed this limit, 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 may want to delete entries after making security changes, to ensure they take effect immediately. Security changes may not take effect immediately when you change a user's rights. For example if you remove a user from a group and a mapping for that user already exists in the WAFL credential cache, the user will continue to have that group's access to files until the entry in the WAFL credential cache times out automatically. The default credential cache timeout period is 20 minutes.

Prerequisites

You must have the name for the entry you want to delete from the WAFL credential cache. To further narrow down the selection, you can optionally specify an IP address.

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 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

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 temporarily retain 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; each instance represents an increasing level of detail.

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

How to solve problems caused by 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 the user's UNIX name or user's Windows 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; each instance represents an increasing level of detail.
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 315 for details about deleting entries.

Determining mapping inconsistencies with a Windows user name

To determine mapping inconsistencies with a Windows user 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 user 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; each instance represents an increasing level of detail.
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 trace CIFS logins by monitoring any attempt by an NFS client to obtain a CIFS credential, complete the following step.

Caution

Use CIFS login tracing carefully because it reports every CIFS login. Persistent use can result in numerous 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.

Tracing domain controller connections

To configure Data ONTAP to send messages to the console when it tries to improve the domain controller connection every few minutes, complete the following step.

Note

Because tracing functions send frequent messages to the console and system log, do not persistently enable this option.

Step	Action
1	Enter the following command: `options cifs.trace_dc_connection on | off` The default is off.

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 326.

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 325.

Caution

SecureShare Access has no Undo command. Use it very carefully.

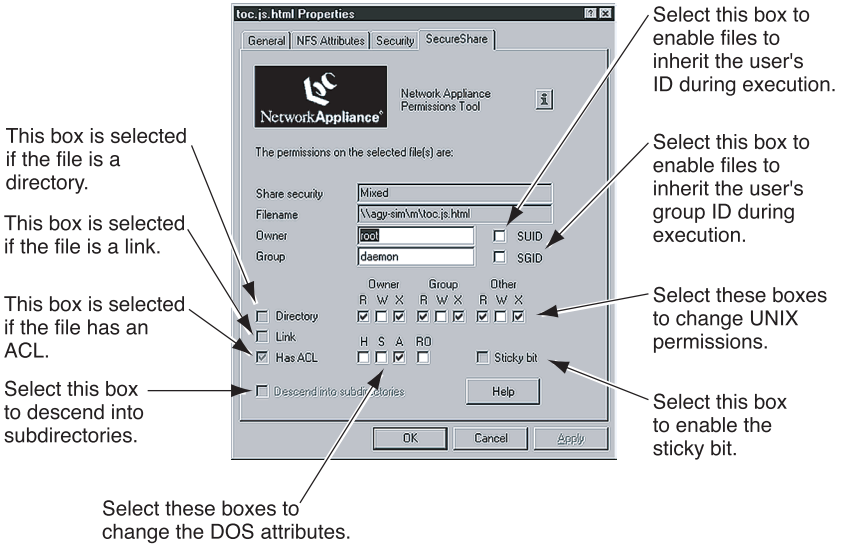
Displaying SecureShare Access permissions information

To display SecureShare Access permissions information, complete the following 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.



Effects of qtree security styles: 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 Windows 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 permissions information" on page 326.
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 that are common to all the items you selected appear with a white background.
- Permissions, DOS attributes, or characteristics that differ 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.
- Clicking a check box until it turns 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 permissions information" on page 326.
2	Change permissions by selecting the appropriate boxes in the SecureShare Access window.
3	Click Apply or OK.

File Access Using DAFS 8

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 330
- "Understanding the DAFS service implementation" on page 332
- "Configuring the DAFS service" on page 333
- "Starting and stopping the DAFS service" on page 337
- "Attaching and detaching network devices" on page 341
- "Understanding session authentication" on page 344
- "Managing exports for DAFS service" on page 349
- "Managing client host access to DAFS exports" on page 354
- "Measuring DAFS performance" on page 359
- "Displaying DAFS status" on page 364
- "Configuring DAFS with options" on page 373

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 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. Network Appliance recommends that you never share files 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 373.
- 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 you can 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 Data ONTAP command reference for `dafs`.

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 VI NIC.

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 cluster failover (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 Data ONTAP command reference for `dafs`.

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; NetApp recommends you use the defaults for this option. For more information about these defaults, see "Default options for the dafs attach command" on page 343.
	`-p` indicates that the VI NIC is attached as a partner interface for cluster failover.
	Result: The DAFS service is attached to the specified VI NIC.
	Example:
	`filer> dafs attach e4`
	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, so it is unnecessary to specify the `-d` option.
	Example for failover VI NIC as a partner interface:
	`filer> dafs attach pe4 -d e4 -p`
	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 an 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 only 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 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, 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.	
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 354.

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 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 Data ONTAP command reference for `dafs export`.

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, you should 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 349.

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 hosts 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.

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.

FPolicy Data ONTAP’s proprietary file policy feature that provides the ability to control access permissions based on file properties, such as file type.

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.

null user — The Windows NT machine account used by applications to access remote data.

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 *See* PCNFS
/etc files. *See* "/etc" entries under E

A

access
  CIFS access to NFS symbolic links, configuring 297
  CIFS files from NFS clients, configuring 313
  CIFS, to NFS directories, optimizing 310
  filer by CIFS client, providing 75
  filer by null session clients, providing 116
  FTP
    anonymous connections, configuring 210
    user connections, configuring 218
  HTTP service 245
  NFS root access, providing 15
  (PC)NFS, configuring 54
  restrictions
    FTP concurrent connections, limiting 212
    FTP user connections, denying 211
    HTTP, directories accessed through URLs, configuring 232
    NFS exported resources, limiting 6
    NFS root access, denying 15
  Windows user account, mapping to UNIX root 90
access control list, see ACLs 53
Account Migrator, SecureShare, desciption of 94
accounts
  migrating, from domain controller to filer 94
  user
    FTP, default anonymous user name 210
    mapping, for UNIX and Windows 79
    UNIX default user 95
    Windows guest 97
ACL characteristics using NFSv4 53
ACLs
  adding, from Server Manager 148
  changing
    from command line 151
    from Server Manager 147
    share-level 143
  displaying, from command line 152
  file-level, description of 153
  for DAFS clients 354
  removing entries
    from command line 152
    from Windows 150
aliases, NetBIOS 69
anonymous FTP access
  configuring 221
  enabling 210
authentication
  configuring, for FTP 218
  for DAFS 345
  for (PC)NFS 54
  method, effects on file system 153

C

caches
  client-side in Windows 2000 and above 103
  for /etc/export entries, checking access 28
  ID-to-name map
    clearing (cifs sidcache clear) 160
  oplocks, description of 180
  SID-to-name map
    changing lifetime of entries in 159
    description of 156
    enabling and disabling 158
  WAFL credential 314
CIFS
  credentials, client UID/GID 75
  default UNIX account, using for guest access 95
  file renaming by clients, preventing 312
  forcing lowercase file naming (options cifs.save_case) 267, 312
  matching users to directories 126
  NFS access to CIFS files, providing 313
  user accounts, copying information to the filer 94

CIFS commands
cifs access
change ACL 151
display ACL 152
remove entries from ACL 152
cifs audit clear (clear internal event log) 174
cifs audit save (update the event log) 168
cifs broadcast (send messages to users) 163
cifs comment (display and change filer description) 68
cifs nbalias (display list of aliases) 70
cifs nbalias load (process entries in cifs_nbalias.cfg) 69
cifs prefdc (manage preferred domain controllers) 200
cifs resetdc (reestablish connection with a domain controller) 202
cifs restart (restart CIFS for a filer) 193
cifs sessions
display information about Windows 2000 clients 184
display share and file information for users 185
display summary session information 184
display user authentication method 119
display user security information 185
cifs setup (reconfigure CIFS) 194
cifs shares
create a share 106
delete a share 110
cifs sidcache clear
clear all cache entries 160
clear cache entries for domain 160
clear specific SID entry 161
clear specific user entry 161
cifs terminate
disable CIFS for a filer 190
disconnect selected client 188
fpolicy disable (disable file screening) 289
fpolicy enable (enable file screening) 290
fpolicy ext excl add (add excluded extensions) 293
fpolicy ext excl remove (remove excluded extensions) 294
fpolicy ext excl reset (reset to default file screening) 294
fpolicy ext excl set (replace list of excluded extensions) 295
fpolicy ext excl show (display list of excluded extensions) 292
fpolicy options default required (require screening for file access) 295
fpolicy servers show default (display screening server status) 296
fpolicy servers stop default (disconnect file screening server) 296
fpolicy show (display file policy information) 291
options cifs.audit.autosave.file.extension counter (specify automatic counter extensions) 172
options cifs.audit.autosave.file.extension timestamp (specify automatic timestamp extensions) 173
options cifs.audit.autosave.file.limit (specify maximum automatic saves) 173
options cifs.audit.autosave.onsize.enable (save automatically by log size) 169
options cifs.audit.autosave.onsize.threshold (specify log size threshold) 169
options cifs.audit.autosave.ontime.interval (specify time interval for automatic saves) 170
options cifs.audit.enable (enable and disable auditing) 166
options cifs.audit.file_access_events.enable (enable and disable file access auditing) 166
options cifs.audit.logon_events.enable (enable and disable auditing of logon and logoff events) 167
options cifs.audit.logsize (specify maximum size of internal event log) 173
options cifs.audit.saveas (save as different event log) 168
options cifs.home_dir
disable home directory shares 142
options cifs.home_dir_namestyle (specify naming style of home directories) 134
options cifs.idle_timeout (specify idle session

timeout) 186
options cifs.netbios_aliases (specify NetBIOS aliases for filer) 69
options cifs.netbios_over_tcp.enable (enable or disable NetBIOS over TCP) 71
options cifs.nfs_root_ignore_acl (allow access to root volume with NTFS security) 196
options cifs.oplocks.enable (enable system-wide oplock) 182
options cifs.oplocks.opendelta (change delay time for sending breaks) 183
options cifs.perm_check_ro_del_ok (allow deletion of files with read-only bit set) 270
options cifs.perm_check_user_gid (set GID to grant access) 145
options cifs.save_case (force lowercase file names) 267
options cifs.shutdown_msg_level (specify which users receive shutdown messages) 192
options cifs.sidcache.enable (enable and disable SID-to-name map cache) 158
options cifs.sidcache.lifetime (change lifetime of mapping entries) 159
options cifs.symlinks.cycleguard (disable access to cyclic directory structures) 309
options cifs.symlinks.enable (enable CIFS clients to follow symbolic links) 298
options cifs.trace_login (trace CIFS logins by NFS clients) 323
options fpolicy.enable (turn file policies on or off) 288
options wafl.default_unix_user (create generic account) 96
CIFS files, accessing, from NFS 313
CIFS protocol
accessing others' home directories 139
accessing root volume with NTFS security (options cifs.nfs_root_ignore_acl) 196
ACLs
adding with Windows administrative tools 148
file-level 153
share-level 143
auditing
clearing audit log 174
description of 164
external log file, /etc/log/adtlog.evt 164
internal log file, /etc/log/cifsaudit.alf 164
list of events 176
logon and logoff events, enabling and disabling (options cifs.audit.logon_events.enable) 167
prerequisites for 164
changing from command line (cifs access) 151
changing share-level ACL 146
CIFS logins by NFS clients, tracing (options cifs.trace_login) 323
cifs setup, reconfiguring 196
cifsconfig.cfg file 194
clearing event audit log (cifs audit clear) 174
cyclic directory structures, disabling access to (options cifs.symlinks.cycleguard) 309
default setting of GID 145
disabling CIFS
for selected clients 187
for the filer (cifs terminate) 190
disconnecting CIFS clients
using command line (cifs terminate) 188
with Windows administrative tools 188
displaying
ACLs, from command line (cifs access) 152
file policy information (fpolicy show) 291
session information 184
share and file information for users (cifs sessions) 185
summary session (cifs sessions) 184
user security information (cifs sessions) 185
event audit log, viewing (Event Viewer) 174
event detail displays
description of 176
from UNIX 178

from Windows 177
lost record 179
unsuccessful file access 178
event log
automatic counter extensions 172
automatic file naming 171
automatic save interval, specifying (options cifs.audit.autosave.ontime.interval) 170
automatic saves, specifying maximum (options cifs.audit.autosave.limit) 173
automatic saving of (options cifs.audit.autosave.onsize.enable) 169
automatic timestamp extensions 172
clearing the internal log (cifs audit clear) 174
counter extensions, specifying (options cifs.audit.autosave.file.extension counter) 172
how event logs are updated 165
internal log, specifying maximum size of (options cifs.audit.logsize) 173
log size threshold, specifying (options cifs.audit.autosave.onsize.threshold) 169
saving as a separate file (options cifs.audit.saveas) 168
size and format of 164
timestamp extensions, specifying (options cifs.audit.autosave.file.extension timestamp) 173
updating (cifs audit save) 168
valid size thresholds 169
valid time intervals 171
file access auditing, enabling and disabling (options cifs.audit.file_access_events.enable) 166
file policies, enabling and disabling (options fpolicy.enable) 288
file screening
disabling (fpolicy disable) 289, 290
enabling (fpolicy enable) 288, 289
excluded extensions list, displaying (fpolicy ext excl show) 292
excluded extensions, adding (fpolicy ext excl add) 293
file policy, creating (fpolicy create) 288
file policy, disabling (fpolicy disable) 290
file policy, enabling (fpolicy enable) 289
included and excluded extensions list, replacing (fpolicy ext exc set, fpolicy ext inc set) 295
included and excluded extensions, removing (fpolicy ext exc remove, fpolicy ext inc remove) 294
included extensions, adding (fpolicy ext inc add) 293
policy, resetting (fpolicy ext exc reset, fpolicy ext inc reset) 294
screening for file access, requiring (fpolicy options default required) 295
screening server status, displaying (fpolicy servers show default) 296
screening servers, adding secondaries (fpolicy options secondary_servers) 296
server, disabling (fpolicy servers stop default) 296
filer description, displaying and changing 68
home directories
accessing 138
creating directories in 135
description of 126
name style, specifying (options cifs.home_dir_namestyle) 134
paths, specifying (options cifs.home_dir) 131
home shares
ceasing to offer (options cifs.home_dir) 142
syntax of, using UNC name 138
idle sessions
description of 186

timeout, specifying (options cifs.idle_timeout) 186
local groups
creating 121
definition of 121
deleting 124
removing members from 124
SnapMirror advisory 121
managing with Windows administrative tools 67
mapping inconsistencies with WAFL credential cache 322
mapping Windows user account for UNIX root access 90
migrating accounts to passwd/group files 94
multiprotocol to NTFS-only, effects of 196
NetBIOS aliases, creating (options cifs.netbios_aliases) 69
NetBIOS over TCP
description of 71
disabling (options cifs.netbios_over_tcp.enable) 71
NTFS-only to multiprotocol, effects of 195
oplocks
changing delay time for sending breaks (options cifs.oplocks.opendelta) 183
delay time for sending breaks, description of 182
description of 180
filer and client oplock settings 181
managing 182
setting, for each qtree (qtree oplocks) 182
system-wide, enabling (options cifs.oplocks.enable) 182
optimizing client access to NFS directory 310
passwd file, creating 92
quotas and generic user account, description of 95
removing ACL entries
from command line (cifs access) 152
from Windows 150
restarting CIFS (cifs restart) 193
sending messages
from command line (cifs broadcast) 163
Server Manager, limitations of 111
shared directories, description of 99
share-level ACLs
reason for changing 146
using GIDs with 144
shares
creating, from command line (cifs shares) 106
deleting, from command line (cifs shares) 110
deleting, from Windows 110
differences between home and other shares 127
forcegroup 100
GID, specifying effect on file access, (options cifs.perm_check_user_gid) 145
group membership 100
number of users, specifying 100
properties, displaying and changing, from Server Manager 107
shutdown messages, specifying recipients of 192
SID-to-name map cache
clearing (cifs sidcache clear) 160
description of 156
enabling and disabling (options cifs.sidcache.enable) 158
lifetime of entries, specifying (options cifs.sidcache.lifetime) 159
starting CIFS at reboot, preventing 191
symbolic links
as home directory names 129
how CIFS clients follow 298
options for accessing 298
UNIX credentials, providing to specific users 76
UNIX guest accounts for CIFS users, creating (options wafl.default_unix_user) 96
UNIX user accounts for CIFS users, creating 93
User Manager, limitations of 111
WAFL mapping inconsistencies 321
Windows guest accounts
creating and deleting 97

cifs setup
configuring WINS servers 66
description of 63
commands, for CIFS *See* CIFS commands
commands, for DAFS *See* DAFS commands
commands, for HTTP *See* HTTP commands
commands, for NFS *See* NFS commands
configuration files
cifs_nbalias.cfg 70
exports file 10
ftpusers 222
group 92
home_dir 131
httpd.access 245
httpd.group 250
httpd.mimetypes 240
httpd.passwd 250
httpd.translations 232
netgroup 21
passwd file 92
symlink.translations 306
usermap.cfg file 81
connections, limiting for FTP 212
credentials, filer access using UNIX 75
cyclic directory structures, description of 309

D

DAFS
client-server implementation 332
information required for setup 333
multiprotocol access to files 332
DAFS authentication
types of authentication sessions 345
types of client access 344
DAFS commands
dafs access (manage client access to DAFS exports) 354
dafs attach (attach DAFS to a VI NIC) 342
dafs detach (detach DAFS from a VI NIC) 343
dafs export (manage DAFS export points) 349
dafs setup (reconfigure DAFS) 334
dafs start (start DAFS on a filer) 337
dafs stats (obtain DAFS statistics) 359
dafs status (display status for DAFS protocol, sessions, and VI NICs) 364
dafs stop (terminate DAFS on a filer) 337
DAFS exports
adding clients to the export's ACL 355
creating 350
destroying 351
disabling 353
displaying ACLs 358
enabling a disabled export 352
listing export points 351
managing ACLs for clients 354
modifying read and write attributes 352
reexporting 353
DAFS performance
client sessions, viewing 370
filer statistics, displaying 360
sessions and requests, displaying 369
state, displaying 365
statistics, obtaining 361
VI NIC sessions, displaying 368
VI NIC status, displaying 366
DAFS protocol
persistent enabling and disabling of 339
starting DAFS on a filer 338
terminating DAFS on a filer 338
transport requirements for DAFS 332
DAFS VI NICs
attaching 342
detaching 343
Data ONTAP
converting to Unicode-formatted directories 311
creating Unicode-formatted directories (options wafl.create_ucode) 310
directory structures
cyclic 309
NFS and CIFS 310
file locking 268
file name creation 267
how usermap.cfg file is used 79
mapping Windows names and UNIX names in 75
matching users to directories using CIFS 126
obtaining GIDs in CIFS 76
obtaining UID in CIFS 75

read-only bit, description of 269
SecureShare Access, description of 325
shared file permissions, description of 269
using NTFS to determine file access 313
using UNIX credentials for file access 75
default UNIX user account
creating (options wafl.default_unix_user) 96
definition of 95
prerequisites for 95
delegating files to clients in NFSv4 48
directory structures
converting to Unicode-formatted directories 311
creating Unicode-formatted directories 310
cyclic directories 309
NFS and CIFS 310
domain controllers
disconnecting and forcing discovery of 202
discovering 198
preferred controllers
adding 200
deleting 200
listing 201
selecting 199
specifying preferred controllers 200
troubleshooting filer connection to 201

E

/etc/cifs_nbalias.cfg file, creating entries in 70
/etc/exports
automatic changes to the export file 9
/etc/exports file
contents of 10
exporting entries differently than specified 29
exporting entries with (exportfs) 27, 28
how it works 9
rules for entries in 10
specified resources in 4
/etc/ftpusers file
about 211
editing 222
format of 222
/etc/group file, finding information about 92
/etc/home_dir file, specifying directories in 131
/etc/httpd.access file
contents of 245
editing 247
format of 247
options in 247
/etc/httpd.group file
contents of 245
creating 250
description of 250
editing 251
/etc/httpd.hostprefixes file 252
/etc/httpd.mimetypes file 240
/etc/httpd.passwd file
contents of 245
creating and editing 250
description of 250
/etc/httpd.translations file
description of 232
fail rule, adding 239
map rule, adding 233
pass rule, adding 237
redirect rule, adding 235
/etc/log/adtlog.evt file (external audit file) 164
/etc/log/cifsaudit.alf file (internal audit file) 164
/etc/log/httpd.log file
changing file format of 259
information in 258
/etc/netgroup file
description of 21
format of entries in 21
/etc/passwd file
contents of 92
format of 92
use of by pcnfsd 54
when created 92
/etc/shadow file, use of by pcnfsd 54
/etc/symlink.translations file (symbolic link translations)
description of 306
listing entries in 308
/etc/usermap.cfg file
contents of 81
format of 80
increasing security in 88
mapping names in 88

use of, by Data ONTAP 79
exporting
access restrictions for exported resources 6
DAFS clients
exporting to 349
reexporting to 353
/etc/exports file 10
exporting files differently than specified in /etc/exports 29
exporting resources to a netgroup 22
matching resources with NFS mount requests 4
resources at different levels of same directory path 5
rules for entries in the /etc/export file 10
rules for exporting resources 5
to a subnet, reasons for 23
ways to export resources 4

F

fail rule, for HTTP 239
FAT (File Allocation Table) file system *See* file system
file access
DAFS notes for multiprotocol filers 332
effects of GID in 145
mapping entries to WAFL credential cache 314
NFS access of CIFS files, description of 313
permissions for NFS and CIFS clients 269
using NTFS to determine 313
file access, determining factors 1
file delegation features, NFSv4 48
file extensions, mapping, for HTTP MIME files 240
file locking
description of 268
NFS operations affected by 268
file names
conventions, in multiprotocol environment 266
creating 267
lowercase option (options cifs.save_case) 267
preventing renaming of 312
file permissions, umask for (PC)NFS clients 57
file screening
enabling 288
file policy
creating 288
deleting 290
disabling 289
displaying 291
enabling 289
exclude list for, adding to 293
include and exclude lists, removing all entries from 294
include and exclude lists, replacing all entries in 295
include list for, adding to 293
included or excluded files, displaying 292
requiring for all files 295
screening servers
disabling 296
secondary screening servers, adding 296
screening servers, displaying status of 296
file system, determining type for a given resource 153
File Transfer Protocol (FTP) *See* FTP service
filers
CIFS description 68
managing CIFS using Windows administrative tools 67
restarting CIFS 193
files, deleting when read-only bit is set 270
FPolicy *See* file screening
FTP commands
ftp stat (provide FTP connection statistics) 223
options ftpd.anonymous.enable (enable or disable anonymous FTP access) 210
options ftpd.anonymous.home_dir (specify home directory for anonymous user) 210, 211, 212
options ftpd.anonymous.name (specify name for anonymous login) 210
options ftpd.auth_style (specify authentication style) 208
options ftpd.enable (enable the FTP daemon) 208
options ftpd.idle_timeout (specify the timeout for idle connections) 212

options ftpd.max_connections (specify maximum concurrent FTP connections) 212
options ftpd.tcp_window_size (specify TCP window size) 212
FTP service
anonymous access
configuring 221
options for 210
authentication styles
Mixed, configuring 220
Mixed, description of 210
NTLM, configuring 220
NTLM, description of 209
overview 208
UNIX, configuring 219
UNIX, description of 208
configuring 214
denying users access to 211
enabling 215
ftpd daemon 208
ftpusers file, format of and editing 211
log files
types of 223
viewing 223
ftp.cmd file, description of 223
ftp.xfer file, description of 223

G

GIDs (group IDs), obtaining 76
groups
how CIFS determines membership in 92
how (PC)NFS determines membership in 54
HTTP service 250
(PC)NFS lookups 54
guest account
FTP anonymous users 210
UNIX default user 95
Windows (default) user, pcuser 97
guidelines, for mapping Windows to UNIX user names 88

H

home directories
accessing others' home directories 139
CIFS
accessing 138
creating 135
description of 126
matching to users 126
naming style, specifying 134
wide symbolic links in 141
FTP, anonymous users 210, 211, 212
home directory paths, CIFS
specifying, in the /etc/home_dir file 131
home shares, ceasing to offer(options cifs.home_dir) 142
HTTP commands
fail (add fail rule to translation file) 239
httpstat (display statistics) 256
httpstat (reset statistics) 257
map (add map to translation file) 233
options httpd.access (restrict access to HTTP services) 243
options httpd.admin.access (restrict access to filer administration via HTTP) 243
pass (add pass rule to translation file) 237
redirect (add redirect rule to translation file) 235
HTTP service
See also "/etc/httpd" entries under E
/etc/httpd.hostprefixes 252
license requirements 229
log file
description of 258
viewing 258
MIME Content-Type values, mapping 240
rules
fail, adding 239
map, adding 233
pass, adding 237
redirect, adding 235
URL translation, filer processing of 232
security
authentication methods 245
basic authentication 245
levels of 242
NTLM authentication 246
restricting access, options for (ifconfig)

243
user password 245
virtual firewall, designating restricted subnets 244
Web pages, protecting 245
setting up 230
starting the HTTP service 230
statistics
description of 254
displaying (httpstat) 256
resetting (httpstat) 257
testing the HTTP service 231
virtual firewall, configuring 244
virtual hosting
description of 252
mapping virtual host addresses (ifconfig) 253
setting up 252

I

ID
obtaining a UID in CIFS 75
obtaining GIDs in CIFS 76
idle sessions
CIFS
description of 186
timeout value, specifying (options cifs.idle_timeout) 186

K

Kerberos
CIFS, configuring for 115
DAFS, configuring for 346
NFS, configuring for 31
keytab generation, NFS 41

L

LDAP
authorizing UNIX client access to NTFS or mixed file systems 278
authorizing Windows client access to UNIX files 277
editing /etc/nsswitch.conf file 275
enabling and disabling (options ldap.enable) 275
servers supported 272
specify base distinguished name (options ldap.base) 272
specify preferred servers (options ldap.servers.preferred) 274
specify servers (options ldap.servers) 274
specifying admin password (options ldap.passwd) 276
specifying admin user name (options ldap.name) 275
specifying LDAP port (options ldap.port) 276
support of Windows client authentication 277
LDAP commands
options ldap.base (specifies base distinguished name) 272
options ldap.enable (enables and disables LDAP) 275
options ldap.name (specifies admin user name) 275
options ldap.passwd (specifies admin password) 276
options ldap.port (specifies LDAP port) 276
options ldap.servers (specifies servers) 274
options ldap.servers.preferred (specifies preferred servers) 274
licenses, HTTP service 229
local user accounts
managing 120
reasons for specifying 119
log files
adtlog.evt 164
cifsaudit.alf 164
ftp.cmd 223
ftp.xfer 223
httpd.log 259

M

Map entries (symbolic link redirects for CIFS clients)
creating 306
description 301
listing 308

requirements for 301
use of 301
map rule for HTTP 233
mapping
inconsistencies with WAFL credential cache 321
SID-to-name cache 156
mapping Windows and UNIX user names
Data ONTAP methods for 79
guidelines for 88
Migrator, SecureShare Account 94
MIME files for HTTP, mapping file suffixes (Content-Type values) 240
mount requests, matched with exported resources 4
multiprotocol environment, file naming conventions 266
multiprotocol filer to NTFS-only filer, effects of 196

N

names
creating, for files in CIFS and multiprotocol environments 267
guidelines for mapping for Windows and UNIX 88
mapping
credentials for Windows and UNIX users 75
mapping for Windows and UNIX with usermap.cfg file 79
NetBIOS aliases
cifs_nbalias.cfg file, processing entries in (cifs nabalias load) 69
creating in /etc/cifs_nbalias.cfg file 70
displaying list of (cifs nbalias) 70
/etc/cifs_nbalias.cfg file 69
NetBIOS over TCP
description of 71
disabling 71
netgroups, configuring 21
network commands
ifconfig (map virtual host addresses) 253
ifconfig (restrict HTTP access over subnet) 244
NFS commands
exportfs (export entries in exports file) 28
option nfs.v4.id_domain (specify user ID domain) 52
options nfs.mountd.trace (enable and disables tracing of denied NFS mount requests) 45
options nfs.vN.enable (enable and disables NFS versions) 46
options pcnfsd (enable and disables pcnfsd) 55
options pcnfsd.umask (define umask) 57
NFS delegation of files (NFSv4) 48
NFS keytab generation 41
NFS protocol
accessing CIFS files, description of 313
CIFS access to NFS directory, optimizing 310
enabling and disabling versions (options nfs.vN.enable) 46
exportfs command
rules for specifying multiple access restrictions 26
targets that can be specified 26
uses for 26
exportfs entry, specifying 10
exporting
description of 10
entries differently than specified in exports file 29
resources to a netgroup 21
resources to subnet 25
exports file, contents of 10
file renaming by CIFS clients, prevention of 312
host restrictions, specifying 13
mapping inconsistencies with WAFL credential cache 321
mount requests
matched with exported resources 4
tracing denied requests 45
netgroups, description of 21
(PC)NFS
clients, authentication of 54
(PC)NFS, creating user entries with 56
pcnfsd daemon
determining group membership 54

enabling and disabling (options pcnfsd) 55
relationship of netgroup and netgroup.byhost map 21
resources clients can mount 4
restrictions based on hosts, specifying 13
restrictions for all hosts, specifying 12
root access
for all clients, allowing 15
for all users, allowing 15
for certain clients, specifying 17
secure NFS access 31
security service
description of 19
specifying 19
symbolic links (absolute), redirecting for CIFS clients 306
symbolic links, using with CIFS 297
umask
definition of (options pcnfsd.umask) 57
(PC)NFS-created files 57
unexporting resources 30
version 4 (NFSv4)
limitations 47
pseudo-fs effect on mount points 52
support 47
user ID domain, specifying (option nfs.v4.id_domain) 52
WAFL mapping inconsistencies 321
WebNFS, configuring 58
NFSv4
ACL characteristics 53
NTFS
converting NTFS filers to multiprotocol filers (wafl.default_security_style option) 195
using to determine file access 313
NTLM, implementation for FTP
description of 209
enabling 220
null sessions, using to access filer data 116

O

oplocks
changing delay time for sending breaks (options cifs.oplocks.opendelta) 183
delay time for sending breaks, description 182
description of 180
filer and client settings 181
managing 182
setting for each qtree (qtree oplocks) 182
system-wide (options cifs.oplocks.enable) 182
when to use 180

P

pass rule for HTTP 237
(PC)NFS
authentication 54
clients, file permissions for 57
creating user entries with 56
group lookups 54
umask
defining (options pcnfsd.umask) 57
working with files 57
pcnfsd daemon
authentication, description of 54
determining group membership 54
enabling and disabling (options pcnfsd) 55
pcuser, UNIX default user for CIFS guest access 95
permissions
DAFS clients permissions 352
shared files permissions 269
umask for (PC)NFS clients 57
UNIX, changing using SecureShare Access 326
preferred domain controller (prefdc), deleting 200
preferred domain controller (prefdc), listing 201
preferred domain controller (prefdc), specifying 200
protocol support 1

Q

qtrees
effects of qtree style on file system 153
setting oplocks for each (qtree oplocks) 182

R

read-only bit

deleting files when set (options cifs.perm_check_ro_del_ok) 270
description of 269
redirect rule for HTTP 235
resources
definition of 3
making available to NFS clients 4
matching NFS mount requests for 4
unexporting of 30
ways to export 4
root
allow access to root volume (CIFS with NTFS security) 196
mapping a Windows user account for UNIX root access 90
root directory, for WebNFS lookup 58
specifying client root access restrictions 15
rules for
entries in /etc/exports file 10
exporting resources 5
HTTP translations file 239
URL translation 232

S

screening policy *See* file screening
SecureShare Access
changing permissions with 326
description of 325
installing 325
SecureShare Account Migrator 94
security
levels for HTTP 242
restricting HTTP access 243
security styles, Kerberos
client support 31, 33
description of 19
implementation 31
setting up for UNIX 38
security styles, specifying, for NFS access 19
Server Manager
ACL
adding entries to 148
displaying and changing 147
removing entries from 150
disconnecting clients 188
limitations of 111
sending messages 162
shares
displaying properties of 107
session information (CIFS), displaying 184
shared directories, description of 99
shared files, permissions for 269
shared resources
with NFS and (PC)NFS clients 3
with WebNFS clients 3
share-level ACLs
changing 146
using GIDs 144
shares
creating
from command line (cifs shares) 106
deleting 110
differences between home and other shares 127
displaying and changing properties 107
forcegroup 100
group membership 100
number of users, specifying 100
share boundary checking for symbolic links
disabling 304
specifying 101
umask value, description 102
virus scanning, specifying 103
wide symbolic links, enabling 101
SID-to-name map cache
changing lifetime of entries 159
clearing (cifs sidcache clear) 160
description of 156
enabling and disabling 158
SNMP, viewing FTP statistics for 224
statistics
DAFS (systat command) 359
FTP (ftp stat) 223
HTTP (httpstat command) 254
NFS file delegation statistics (nfsstat command) 50
WAFL credential cache 318
symbolic links
CIFS home directories, using symbolic links

for 129
description of 297
disabling share boundary checking for 304
enabling CIFS clients to follow (options cifs.symlinks.enable) 298
preventing links to files 300
redirecting
for CIFS clients 306
methods of 306
using Map entries 301
using Widelink entries 302
wide symbolic links, enabling for a share 101

T

timeout value, specifying for FTP 212
translations
for HTTP directories 232
of symbolic links 306
rules for URL 232

U

UID, obtaining 75
umask value, description 102
umask, defining for (PC)NFS 57
UNC name, syntax for specifying home share 138
unexporting resources 30
Unicode directories, converting to 311
Unicode format, directory structure 310
UNIX
credentials, description of 75
permissions, changing 326
user name, special characters in 72
using UNIX credentials for filer access 75
URLs
redirecting and restricting access to directories 232
translation rules 232
user accounts
copying information from a domain controller to the filer 94
FTP anonymous user 210
(PC)NFS creating entries for 56
UNIX default user 95
Windows guest 97
user authentication method, displaying 119
user ID (UID), obtaining 75
User Manager
creating a local group 121
deleting group 124
limitations of 111
removing members from group 124
user names
mapping Windows and UNIX 75
matching to directories using CIFS 126
usermap.cfg, description of 79

V

virtual firewall for HTTP 244
virtual hosting for HTTP 252
virus scanning, specifying for a share 103

W

WAFL credential cache
adding entries to (wcc) 314
deleting entries in (wcc) 315
description of 313
displaying cache statistics (wcc) 318
mapping inconsistencies
description of 321
with a CIFS name 322
with a UNIX name 321
monitoring 318
setting length of validity (options wafl.wcc_minutes_valid) 317
WAFL, setting security style option (wafl.default_security_style option) 195
Web page protection 245
WebDAV (Web-based Distributed Authoring and Versioning), described 261
WebNFS
configuring 58
root directory 58
wide symbolic links
enabling for a share 101
in home directories 141
Widelink entries
creating 307
description 302

how to list 308
how used 302
limitations 302
requirements 302
window size, specifying for FTP TCP 212
Windows and UNIX user names (usermap.cfg file)
mapping by DataONTAP 79
Windows and UNIX user names, guidelines for (custom) mapping 88
Windows domain names, Data ONTAP interpretation of 84
Windows NT domain names, Data ONTAP interpretation of 84
Windows user names
converting from Windows 2000 format to pre–Windows 2000 format 72
WINS server addresses, changing 194
WINS servers, configuring 66