Data ONTAP™ 6.4
System Administration
Data Protection 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 210-00325
January 2003

Copyright and trademark information

Copyright information

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

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

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

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

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

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

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

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

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

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

1. Redistributions of source code must retain the above copyright notices, this list of conditions, and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

   This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

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

IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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

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

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

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

Trademark information

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

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

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

Table of Contents

Preface xiii

Chapter 1 **Introduction to Data Protection** 1

Data protection options 2

Online backup and recovery 6

Tape backup and recovery 9

Database protection with the NVFAIL feature 11

Deciding which data protection method to use 12

Protecting against a data loss disaster 14

Data protection in a SAN environment 18

Chapter 2 **Snapshot Management** 19

Understanding snapshots 20

Accessing snapshots from clients 22

Restoring files from snapshots 26

Creating snapshot schedules 27

Creating snapshots 33

Finding the snapshot you need 34

Understanding snapshot disk consumption 37

- Monitoring snapshot disk consumption 40
- Displaying snapshot disk consumption statistics 41
- Understanding the snapshot reserve 45
- Changing the snapshot reserve 48

Saving disk space by using file folding 49

Deleting snapshots 50

Renaming snapshots 53

Chapter 3 **Data Reversion Using SnapRestore** 55

Understanding SnapRestore 56

Deciding when to use SnapRestore 57

Installing the SnapRestore license 60
Reverting a volume to a selected snapshot 61
Reverting a file to a selected snapshot 65
Obtaining correct incremental backups after reversion. 70

Chapter 4 **Data Protection Using SnapMirror** 71
SnapMirror overview 72
SnapMirror commands and configuration files. 79
Considerations when planning and running SnapMirror 83
Setting up a basic SnapMirror operation 86
Specifying destination filers on the source filer 89
Defining source and destination through snapmirror.conf 92
Enabling SnapMirror 100
Initializing a SnapMirror destination 102
Updating a destination manually 110
Listing SnapMirror snapshots 113
Checking SnapMirror data transfer status 116
Checking SnapMirror data transfer logs 125
Aborting a SnapMirror transfer. 130
Stabilizing (quiescing) destinations before taking a snapshot 132
Turning off or changing scheduled updates for volumes or qtrees. 135
Turning off SnapMirror updates for the filer 137
Converting a destination to a writable volume or qtree 139
Releasing partners from a SnapMirror relationship 142
Resynchronizing SnapMirror. 144
Copying from one destination to another in a series (cascading) 151
SnapMirroring a volume to local tape 156
How SnapMirror works with the dump command 167
Fixing changes to SnapMirror elements 168

Chapter 5 **Data Protection Using SnapVault** 171

SnapVault overview 172

Planning SnapVault. 180

Setting up a basic SnapVault backup for filer systems 183

Setting up SnapVault backup on Open Systems platforms. 190

Managing SnapVault backup of Open Systems platforms 196

Enabling SnapVault. 201

Starting a SnapVault backup relationship 205

Scheduling SnapVault snapshot updates 209

Checking SnapVault transfers 214

Displaying SnapVault snapshots 220

Changing settings for SnapVault backup relationships. 224

Manually updating individual secondary storage system qtrees 226

Creating a snapshot manually 228

Restoring SnapVault data to the primary storage system 230

Aborting SnapVault transfers. 235

Ending SnapVault backups for a qtree 237

Unscheduling SnapVault snapshots 238

Releasing SnapVault relationships 240

Protecting your SnapVault backups through SnapMirror 241

Turning SnapVault off 244

Chapter 6 **Volume Copy** 245

Learning about volume copy 246

Preparing to copy a volume. 248

Verifying the size of each volume 250

Verifying the relationship between filers 251

Verifying and changing status of source and destination volumes . . .252

Enabling remote access 254

Copying volumes 255

Checking the status of a volume copy operation 260

Displaying the current speed for copying a volume 262

Controlling the speed of a volume copy operation 263
Aborting a volume copy operation 265

Chapter 7 **Mirrored Volume Management Using SyncMirror** 267
Understanding mirrored volumes 268
Enabling and disabling the mirroring license 271
Creating mirrored volumes 272
Creating a mirrored volume when creating a volume 275
Adding a plex to an existing unmirrored volume 279
Adding disks to mirrored volumes 282
Changing the state of a plex 286
Splitting mirrored volumes 288
Removing and destroying a plex from a mirrored volume 290
Comparing the plexes of mirrored volumes 291

Chapter 8 **Tape Drive Management** 295
Understanding tape devices 296
Displaying tape device statistics 299
Displaying supported tape drives 301
Assigning tape aliases 303
Displaying tape device and tape drive information 311
Understanding tape files 315
Controlling tape drives 316
Qualifying tape drives dynamically 321
Using nonqualified tape drives 325
Enabling and disabling persistent reservations 329

Chapter 9 **Data Backup to Tape** 331
Introduction to data backup to tape 332
Types and methods of backup 333
What the dump command backs up 335
How the dump command works 336

Preparing to use the dump command 338
Determining the amount of backup data 339
Estimating the number of tapes for the backup 340
Deciding how to enter the dump command 341

Using the dump command 342
Dump command syntax 343
Recommendations for performing a backup 346
Specifying the backup level 348
Understanding increment chains 349
Specifying the tape device 351
Specifying the dump path 353
Specifying a list of files 355
Excluding items from a backup 356
Specifying a name for a backup 358
Specifying no Windows NT ACLs. 359
Specifying the blocking factor 360
Specifying the tape file size 362
Specifying updates to the /etc/dumpdates file 363

Appending backups to a tape 365

Examples of the dump command. 366
Level-0 backups to a local tape drive 367
Incremental backups to a local tape drive 369
Backups to a remote tape drive. 370
Backups to multiple tapes 372
Backup of a directory from a snapshot. 374
Backup of a list of files or directories 375
Backup data without ACLs. 376
Backup specifying a blocking factor 377
Backup excluding specified files. 378

Verifying the contents of a dump command backup 379

Checking the status of a dump command backup 380

Restarting a dump command backup 382

Deleting restartable dump command backups 384

Chapter 10 **Data Restoration from Tape** 385

About data restoration 386

What the restore command recovers 387

How the restore command works. 388

Preparing to use the restore command 389

Getting required materials 390
Deciding where to enter the restore command 391
Preparing the destination 392

Using the restore command 393
Restore command principles 394
Specifying an individual file restore 398
Specifying a full restore 399
Specifying a table-of-contents restore 400
Specifying a resumed restore 402
Specifying tape devices 404
Specifying a single tape file on a multifile tape 406
Specifying the restore destination 407
Specifying the blocking factor 408
Specifying detailed status output 409
Specifying ignoring inode limitations 410
Specifying automatic confirmations 412
Specifying no Windows NT ACLs 413
Specifying no qtree information 414
Specifying a test restore 415

Restoring incremental backups 416

Restoring the entire filer 417

Restore examples 418
Restoring an entire backup 419
Restoring individual files or directories 421
Restoring using a remote tape drive 422
Multiple tape restores 423

Chapter 11 **Using NDMP Services** 425

Learning about NDMP services on the filer 426

Using tape devices with NDMP 429

Preparing a filer for basic NDMP backup application management 431

Enabling or disabling NDMP service 433

Managing NDMP security features 434

Specifying the NDMP version 439

Specifying a preferred network interface 441

Displaying NDMP session status 443

Displaying NDMP debug messages 446

Terminating an NDMP session 449
Copying data with ndmpcopy 450

Chapter 12 **Event Logging for Backup and Restore** 455
About event logging for backup and restore 456
Event log messages 457
Enabling or disabling event logging 462

Chapter 13 **Database Protection Using NVFAIL** 463
Understanding the nvfail option 464
Enabling and disabling database file protection 466
Using the nvfail_rename file for additional database protection 467

Chapter 14 **Virus Protection for CIFS** 469
Understanding CIFS virus protection. 470
Setting up and starting virus scanning 472
Specifying file types to be scanned. 475
Excluding file types to be scanned 477
Specifying shares for scanning 481
Turning scanning on or off for shares 482
Turning scanning on or off for read-only access for shares 484
Adding shares with virus scanning turned on or off 485
Displaying the scanner list 487
Checking vscan information 488
Setting and resetting the virus scan request timeout 490
Allowing file access when the scan cannot be performed 491
Controlling vfiler usage of host filer virus scanners 492
Checking the status of virus-scanning options 493
Stopping a virus scanner session 494
Resetting the scanned files cache. 495
Enabling virus scan messages to CIFS clients 496

Appendix A **Disaster Protection Using MetroCluster** 497

Configuring disaster recovery using MetroCluster 498
MetroCluster configuration up to 500 meters 499
MetroCluster configuration for distances up to 10 kilometers 501

Recognizing a disaster 503

Recovering from a disaster 505

Glossary 513

Index 519

Preface

About this guide

This guide describes how to protect, back up, restore, and copy data between Network Appliance™ filers that run Data ONTAP™ 6.4 software and that are serving Data by Design™. This guide covers all filer models.

Audience

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

Terminology

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

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

Command conventions

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

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

Keyboard conventions

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

Typographic conventions

The following table describes typographic conventions used in this guide.

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

Special messages

This guide contains special messages that are described as follows:

Note

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

Caution

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

WARNING

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

Introduction to Data Protection 1

About this chapter

This chapter introduces the data backup, restore, and protection capabilities that are available in this release of Data ONTAP.

Topics in this chapter

This chapter discusses the following topics:

- "Data protection options" on page 2
- "Online backup and recovery" on page 6
- "Tape backup and recovery" on page 9
- "Database protection with the NVFAIL feature" on page 11
- "Deciding which data protection method to use" on page 12
- "Protecting against a data loss disaster" on page 14
- "Data protection in a SAN environment" on page 18

Data protection options

About data protection

Data protection means backing up data and being able to recover it. You protect the data by making copies of it so that it is available for restoration even if the original is no longer available.

Why data needs protection

Reasons that businesses need data backup and protection systems include the following:

- To protect data from accidentally deleted files, application crashes, data corruption, and viruses
- To archive data for future use
- To recover from a disaster

Methods of protecting data

Depending on your data protection and backup needs, Data ONTAP offers a variety of features and methods to ensure against accidental, malicious, or disaster-induced loss of filer data.

Data Protection Feature	Description
SnapShot™	Backup within a volume. This feature allows you to manually or automatically create, schedule, and maintain multiple backups (also called snapshots) of data on a volume. Snapshots use only a minimal amount of additional volume space on your filer and do not have a performance cost. If a user accidentally modifies or deletes crucial data on a volume with SnapShot enabled, that data can be easily and quickly restored from one of the last several snapshots taken. See Chapter 2, "Snapshot Management," on page 19.

Data Protection Feature	Description
SnapRestore™ (license required)	Fast, space efficient restoration of large volumes of data backed up to snapshots. The SnapRestore feature performs on request snapshot recovery from snapshots on an entire volume. See Chapter 3, "Data Reversion Using SnapRestore," on page 55.
SnapMirror® (license required)	Volume-to-volume and qtree-to-qtree replication. This feature enables you to periodically make snapshots of data on one volume or qtree, replicate that data to a partner volume or qtree, usually on another filer, and archive one or more iterations of that data as snapshots. Replication on the partner volume or qtree ensures quick availability and restoration of data, from the point of the last snapshot, should the filer of the original volume or qtree be disabled. If you conduct tape backup and archival operations, you can carry them out on the data already backed to the SnapMirror partner filer, thus freeing the original filer of this time-consuming, performance-degrading chore. See Chapter 4, "Data Protection Using SnapMirror," on page 71.

Data Protection Feature	Description
SnapVault™ (license required)	Centralized backup of multiple qtrees on multiple filers using snapshot technology. This feature enables you to back up qtrees on multiple volumes and filers to a single Network Appliance SnapVault secondary storage system specialized for quick backup and restore of its filer sources. You can also install the *SnapVault Open Systems* agent on non-filer Windows NT, Windows 2000, Solaris, or HP-UX systems. This agent allows SnapVault to back up and restore data to these systems also. If you conduct tape backup and archival operations, you can carry them out on the data already backed up to the SnapVault secondary storage system, thus freeing your filers of this time-consuming, performance-degrading chore. See Chapter 5, "Data Protection Using SnapVault," on page 171.
Tape backup `dump` and `restore` commands	Tape backup and restore. The `dump` and `restore` commands allow you to back up filer snapshots to tape. The `dump` command takes a snapshot of the volume and then copies that data to tape. Because the snapshot, not the active file system, is backed up to tape, the filer can continue its normal functions while the tape backup takes place. See Chapter 8, "Tape Drive Management," on page 295, Chapter 9, "Data Backup to Tape," on page 331, and Chapter 10, "Data Restoration from Tape," on page 385.
vol copy	Fast block-copy of data from one volume to another. The `vol copy` command enables you to quickly block-copy stored filer data from one volume to another. See Chapter 6, "Volume Copy," on page 245.

Data Protection Feature	Description
SyncMirror™ (cluster configuration required)	Continuous mirroring of data to two separate filer volumes. This feature allows you to mirror filer data real-time to matching volumes physically connected to the same filer head. In case of unrecoverable disk error on one volume, the filer automatically switches access to the mirrored volume. Filer cluster configuration is required for this feature. See Chapter 7, "Mirrored Volume Management Using SyncMirror," on page 267.
NDMP support	Support for tape backup management of the filer by third-party applications that use the Network Data Management Protocol (NDMP). Data ONTAP also provides an NDMP copy command that you can use to carry out NDMP-compliant backups and restores. Security login restricts access to NDMP operations. See Chapter 11, "Using NDMP Services," on page 425.
nvfail option	Protection against data corruption by failures of non-volatile RAM (NVRAM). See Chapter 13, "Database Protection Using NVFAIL," on page 463.
virus scan support	Support for third-party virus-scanning software for files accessed by CIFS clients. See Chapter 14, "Virus Protection for CIFS," on page 469.
MetroCluster	SyncMirror functionality enhanced to provide continuous volume mirroring over 500-meter to 10-kilometer distances. See Appendix A, "Disaster Protection Using MetroCluster," on page 497.

Online backup and recovery

About online backup and recovery

Data ONTAP creates online data backups to enable online data recovery. Online backup refers to backup data that is stored on disks rather than on tape. Data stored on disk is thus available for quick restoring in event that disaster recovery operations are necessary.

Online backup and recovery solutions include: SnapShot, SnapMirror, SnapRestore, SnapVault, SyncMirror and the `vol copy` command.

- The SnapShot feature enables you to schedule weekly, daily, or hourly online backups. Snapshot technology makes online point-in-time copies in the same volume as the original data and enables users to recover their own deleted or modified files without assistance from a system administrator.
- The SnapRestore feature restores an entire volume to the state recorded in a previously created snapshot with maximum speed and disk space efficiency.
- The SnapMirror feature performs regularly scheduled automated file system snapshot replication of a volume or qtree onto another volume or qtree on the same or a separate filer.
- The SnapVault feature protects the data in one or more qtrees in a series of snapshots stored on a separate filer. SnapVault maintains an online, asynchronous, permanently read-only replica of the qtree data. SnapVault backup and snapshot creation runs on an automated schedule.

  Note

  SnapVault, in addition to providing backup to NetApp filers, also provides direct backup of non-filer storage systems, such as servers running Windows NT, Windows 2000, Solaris, or HP-UX.

- SyncMirror provides continuous real-time mirroring of data between two partner volumes on a shared or partner filer system.
- You can also use the `vol copy` command to snapshot and copy file systems to separate volumes or filers manually or by means of a script. You can use the `ndmpcopy` command to copy any subtree to any location on any filer.
- The MetroCluster feature provides SyncMirror continuous mirroring over extended distances (500 meters to 10 kilometers).

You can use these online data backup and recovery systems to supplement tape backup and recovery.

Advantages of online backup and recovery

Online backup and recovery protection gives you the following advantages over tape archives alone:

- Speedy backups and restores greatly reduce backup time requirements.
- Backups can be made more frequently because they are faster.
- It is easy to recover a particular file, directory, or volume from an online backup.
- Disaster recovery is quicker with online mirroring and restores.
- Data availability is higher because of the high speed of data recovery.
- More data can be backed up in less time.

Disadvantages of online backup and recovery

Online snapshot data protection has the following disadvantages over tape archives:

- Online data protection is physically vulnerable. Filers and disk shelves are vulnerable to physical catastrophes.
- Online data protection consumes resources, such as disk space, that could be used for day-to-day activities.

Online backup and restore methods

To find detailed information about Data ONTAP data protection and online backup and recovery methods, consult the following table.

Method	Chapter
Using snapshots to make a read-only image of a file system on the same disk	Chapter 2, “Snapshot Management,” on page 19
Using SnapRestore to restore data to a corrupted volume from a previous snapshot	Chapter 3, “Data Reversion Using SnapRestore,” on page 55
Using SnapMirror to maintain a replica of one volume in another volume	Chapter 4, “Data Protection Using SnapMirror,” on page 71

Method	Chapter
Using SnapVault to keep copies of volumes on the server, from which individual qtrees are available at any time to the client	Chapter 5, "Data Protection Using SnapVault," on page 171
Using the `vol copy` command to copy data from one volume to another	Chapter 6, "Volume Copy," on page 245
Using SyncMirror to maintain two identical copies of a volume at all times	Chapter 7, "Mirrored Volume Management Using SyncMirror," on page 267

Tape backup and recovery

When to use tape backup and recovery

You use tape backup and recovery to create tape archives or retrieve data from tape archives. Tape archives complement online backup methods, and you should plan to do tape backups regardless of the online backup and recovery methods you use.

Advantages of tape archives

Making tape archives gives you the following advantages over relying solely on online storage:

- Once made, they require few resources to maintain.
- You can put the archives in a more secure place than you can put a filer.
- You can recover data on Sun Solaris systems with the `ufsrestore` command.
- Data archived on tape is on a physically stable medium and is stored in a standard format. You can recover data from any release of Data ONTAP.

Disadvantages of tape archives

Tape archives have the following disadvantages compared to online storage:

- It takes a long time to restore data from tape.
- It is time-consuming to find a particular file or directory on tape.
- Some systems are too large to be backed up to tape efficiently.

  Large file systems might need to be broken into multiple volumes and backed up on different nights. Volumes over 500 GB or total data set sizes greater than 1.4 TB can take a long time to restore. These volumes are good candidates for backup using the SnapMirror or SnapVault features.

Where to find backup and restore information

If you want to perform backups to tape and restores from tape, you should be familiar with the topics listed in the following table.

Topic	Chapter
Understanding tapes and controlling tape drives	Chapter 8, “Tape Drive Management,” on page 295

Topic	Chapter
Backing up data to tape	Chapter 9, “Data Backup to Tape,” on page 331
Restoring data from tape	Chapter 10, “Data Restoration from Tape,” on page 385
Using the NDMP protocol, which third-party backup software vendors use	Chapter 11, “Using NDMP Services,” on page 425

Database protection with the NVFAIL feature

About database protection

If NVRAM problems occur that compromise database validity, the NVFAIL feature can warn you and automatically rename the database so that it does not restart automatically. You can then make sure that the database is valid before restarting it.

Where to find more information

See "Database Protection Using NVFAIL" on page 463 for more information about how Data ONTAP provides database protection using the `nvfail` option of the `vol options` command.

When to use NVFAIL

You use this feature only when you have databases on your filer.

Deciding which data protection method to use

Evaluating types of data protection

You can store data in tape archives, on filers, or both. Although tape archives can be cumbersome for data recovery, they are essential and complement online data protection. Therefore, you might want to use multiple data protection capabilities.

Data protection table

The following table summarizes the data protection features that are best for specific tasks.

Task	Sync-Mirror	Snap-shots	Vol-copy	Snap-Mirror	Snap-Restore	Snap-Vault	ndmp copy	Tape backup and recovery	NV-FAIL
Archival backup and recovery	-	-	-	X	X	X	-	X	-
Mirroring a volume or qtree on another filer	-	-	-	X	-	-	-	-	-
Centralized backup of multiple filer and non-filer storage systems	-	-	-	-	-	X	-	-	-
Moving a volume or qtree among filers	-	-	X	X	-	-	X	-	-

Task	Sync-Mirror	Snap-shots	Vol-copy	Snap-Mirror	Snap-Restore	Snap-Vault	ndmp copy	Tape backup and recovery	NV-FAIL
Quick recovery of deleted items	-	X	-	X	X	X	-	-	-
Rearranging data within a filer	-	-	X	X	-	X	X	-	-
Recovery from double-disk failure	X	-	X	X	-	X	X	X	-
Reverting a filer to a previous state	-	-	-	-	X	X	-	-	-
Protecting databases against NVRAM errors	-	-	-	-	-	-	-	-	X

Protecting against a data loss disaster

What a data loss disaster is

A disaster is a situation in which service from one physical site (for example, a building or a corporate campus) on the network is lost for an extended period of time. The following are examples of disasters:

- Fire
- Earthquake
- Prolonged power outages at a site
- Prolonged loss of connectivity from clients to the filers at a site

What a disaster affects

When a disaster occurs, it can affect all of your computing infrastructure including Network Appliance filers, application servers, networking connectivity, and client connectivity. When you create a disaster plan, take into consideration all of your computing infrastructure.

Determining if a disaster occurred

It is critical that you follow some predefined methodical procedure to confirm that a disaster really has occurred. The procedure should determine the status of the supposed disaster site.

- Use external interfaces to the filer, such as the following:
  - Ping
  - Remote shell
  - FilerView
- Use network management tools to test connectivity to the disaster site.
- Physically inspect the disaster site, if possible.

You should declare a disaster only after determining that a disaster has occurred and that service cannot be restored.

Tools for protecting against data loss disasters

The following Network Appliance features and products are best for enabling the administrator to back up or replicate data stored on a local filer to an off-site network location. This ensures data can be restored if data loss is caused by disaster at a primary data storage site.

Feature	Description
SnapVault: Inter-site snapshot backup and restorability.	A SnapVault secondary storage device can be located offsite, any distance from the primary storage units that it is backing up. **Data recoverability:** In event of a data-loss disaster at a primary storage site, data that is backed up to SnapVault secondary storage can be restored to primary storage units that have not suffered physical damage or that have been replaced either at the disaster site or at an alternate location. **Currency of restore:** Data can be restored from the time that the last SnapVault snapshot was created on the secondary storage system. **Connection requirements:** DSL connections or faster are recommended between the primary and secondary storage systems. Modem connections are possible. Routers, switches, and DNS servers should be pre-configured to direct users to alternate primary storage sites if the primary storage system that they first attempt to access becomes unavailable. **Advantage:** Centralized, inexpensive off-site backup **For more information:** See Chapter 5, "Data Protection Using SnapVault," on page 171.

Feature	Description
SnapMirror: Inter-site snapshot backup, availability, and restorability	A SnapMirror destination storage device can be located offsite, any distance from the source filer whose volumes it is mirroring. **Data availability:** In event of a data-loss disaster at a source filer site, data that has been snapmirrored to the destination site can be made quickly available at the destination site. **Data recoverability:** SnapMirror data can be restored to source storage units that have not suffered physical damage or that have been replaced either at the disaster site or at an alternate location. **Currency of restore:** Data can be restored from the time of the last SnapMirror snapshot transfer from source to destination. **Connection requirements:** DSL connections or faster are recommended between the source and destination filers. Modem connections are possible. Routers, switches, and DNS servers should be pre-configured to direct users to the destination storage site if the source filer they are attempting to access becomes unavailable. **Advantage:** Combined off-site protection and availability **For more information:** See Chapter 4, “Data Protection Using SnapMirror,” on page 71.

Feature	Description
Metro-Cluster: Inter-site real-time backup, availability, and restorability	Synchronously mirrored MetroCluster filers can be located at different sites, up to ten miles from one another. **Data availability:** In event of a data-loss disaster at one filer site, data that has been mirrored to the partner filer site can be made quickly available. **Data recoverability:** This data can also be mirrored to source storage units that have not suffered physical damage or that have been replaced either at the disaster site or at an alternate location. **Currency of restore:** Data can be restored from the time of the last NVRAM checkpoint. **Connection requirements:** Data ONTAP cluster connections supplemented with switches and DSL or faster connections are required. Routers, switches, and DNS servers should be pre-configured to direct users to the MetroCluster partner if the clustered filer that they first attempt to access becomes unavailable. **Advantage:** Combined real-time off-site protection and availability **For more information:** See Appendix A, "Disaster Protection Using MetroCluster," on page 497.

Data protection in a SAN environment

Protecting volumes containing LUNs

If your filer volumes contain logical units of storage (LUNs) created to accommodate integration into a NetApp storage area network (SAN) environment, you must carry out modified procedures to implement data protection using the features described in this document.

See the chapter on data protection in the *SAN System Administrator's Guide* for description of data backup and restore on filer volumes containing LUNs.

Snapshot Management 2

About this chapter

This chapter discusses what snapshots are and how to manage them.

Topics in this chapter

This chapter discusses the following topics:

- "Understanding snapshots" on page 20
- "Accessing snapshots from clients" on page 22
- "Restoring files from snapshots" on page 26
- "Creating snapshot schedules" on page 27
- "Creating snapshots" on page 33
- "Finding the snapshot you need" on page 34
- "Understanding snapshot disk consumption" on page 37
- "Saving disk space by using file folding" on page 49
- "Deleting snapshots" on page 50
- "Renaming snapshots" on page 53

Understanding snapshots

What a snapshot is

A snapshot is a frozen, read-only image of the entire Data ONTAP file system that reflects the state of the file system at the time the snapshot was created. Snapshots are your first line of defense for backing up and restoring data.

When snapshots are created

Data ONTAP maintains a configurable snapshot schedule that creates and deletes snapshots automatically. See "Creating snapshot schedules" on page 27. Snapshots can also be created and deleted manually.

Maximum number of snapshots

You can store up to 255 snapshots at one time on each filer volume.

Maximum space snapshots can occupy

You can specify the percentage of disk space that snapshots can occupy. The default setting is 20% of the total (both used and unused) space on the disk. For a full explanation of how snapshots consume disk space, see "Understanding snapshot disk consumption" on page 37.

How snapshots handle file permissions

Snapshot files carry the same permissions and inode numbers as the original files, keeping the integrity of the security system intact. Inodes are data structures that hold information about files on the filer. There is an inode for each file and a file is uniquely identified by the file system on which it resides and its inode number on that system.

Note

The inode number for a file in a snapshot is the same as the inode number for the corresponding file in the active file system. As a result, some programs on UNIX clients consider the two files to be the same. For example, if you use the GNU `diff` program to compare the two files, it does not find any differences between them. In some cases, if you try to restore a file from a snapshot, you might see the following error message:

```
cp:.snapshot/xxx and xxx are identical.
```

To make sure that the two files have different inode numbers before the copy or comparison, copy one of the files to different name.

What you can do with snapshots

Snapshots enable system administrators to create instantaneous backups from an active filer.

Snapshots enable end-users to

- Recover older versions or sets of files that were accidentally changed or deleted
- Restore their own files without needing a system administrator to restore files from tape

Accessing snapshots from clients

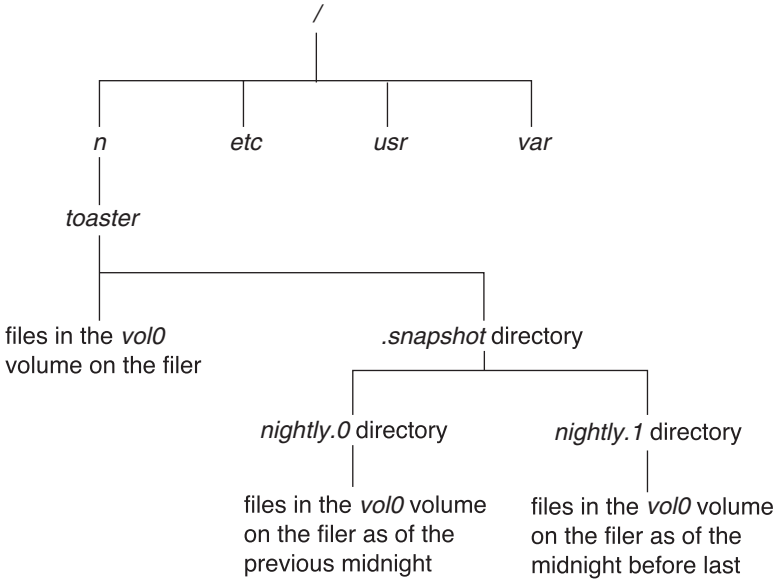
About user access to snapshots

By default, every directory in the filer's active file system contains a directory named .snapshot, through which users can access old versions of files in that directory. How users gain access to snapshots depends on the file-sharing protocol used—NFS or CIFS—and is described in the following sections. Access to snapshots can be turned off.

Snapshot files carry the same permissions as the original file. A user who has permission to read a file in the active file system can read that file in a snapshot. A user without read permission to the active file cannot read that file in a snapshot. Write permission does not apply because snapshots are read-only. Snapshots can be accessed by any user with the appropriate permissions.

NFS user access to snapshots

The following illustration shows the directory structure on an NFS client with the vol0 volume of a filer named toaster mounted on the /n/toaster directory.



Explanation: In this example, the user can obtain access to snapshots by way of the /n/toaster/.snapshot directory. Notice that the .snapshot directory is shown only at the mount point, although it actually exists in every directory in the tree.

The user, however, can only see the .snapshot directory at the mount point. That is, the .snapshot directory is accessible by name in each directory, but is only seen in the output of the `ls` command at the mount point.

For example, at the mount point of a filer file system, a directory listing looks like this:

```
filerA> ls -a
.     ..     .snapshot     dir1     dir2
```

The same command entered in a directory below the mount point does not show the .snapshot directory; for example:

```
filerA> cd dir1
filerA> ls -a
.     ..    file1     file2
```

If you enter the `ls` command with the directory name .snapshot, you can see a directory for each of the snapshots for the dir1 directory:

```
filerA> ls .snapshot
hourly.0     hourly.4     nightly.0     nightly.4
hourly.1     hourly.5     nightly.1     nightly.5
hourly.2     hourly.6     nightly.2     weekly.0
hourly.3     hourly.7     nightly.3     weekly.1
```

If the .snapshot directory entry appeared in every directory, it would cause many commands to work improperly. For instance, all recursive commands for deleting files would fail because everything below the .snapshot directory is read-only. Recursive copies would copy everything in the snapshots as well as files in the active file system, and a `find` command would generate a list much longer than expected.

CIFS user access to snapshots

To CIFS users, the .snapshot directory appears only at the root of a share. For example, if a user's home directory is a share named bill that corresponds to the /vol/vol0/home/bill directory, only the /vol/vol0/home/bill/.snapshot directory is visible. When this user displays the contents of the home directory, the .snapshot directory is displayed as ~snapshot if the operating system supports long file names and as ~SNAPSHT if the operating system supports only short file names.

Note

The .snapshot directory can be made visible in a directory listing or Windows Explorer display if the client operating system is configured to show hidden files.

In each directory within the share, a snapshot directory exists but is not visible to clients. For example, if the client operating system supports long file names, the applications on that operating system can use the snapshot at each level of the share by using .snapshot, ~snapshot, or ~SNAPSHT as the directory name. The user cannot, however, display the directory name in any listing.

Accessing snapshots from CIFS clients

To access snapshots on Windows NT 4 or other Windows® clients (Windows 95 or later), complete the following step.

Step	Action
1	From the Start > Run menu, enter the following command: **`\\filername\share\.snapshot`** (or **`~snapshot`** or **`~SNAPSHT`**). *filername* is the name of the filer you are using. *share* is the name of the share you want to access.

Example: `\\filerA\home\.snapshot`

Snapshots can also be accessed lower in the share by providing a path to a lower directory. Snapshots can be accessed through DOS on any system by changing to the ~SNAPSHT directory.

Disabling and enabling client access to snapshots

Sometimes you might want to disable and enable client access to snapshots in a specific volume, for example, for security reasons or to prevent access to corrupted or virus-infected files.

To disable or enable client access to snapshots in a volume, complete the following step.

Step	Action	
	If you want to...	**Then...**
1	Disable client access to snapshots and make the .snapshot directory invisible to clients	Enter the following command: **`vol options volume_name nosnapdir on`** *volume_name* is the name of the volume for which you want to disable client snapshot access.
	Enable client access to snapshots and make the .snapshot directory visible to clients	Enter the following command: **`vol options volume_name nosnapdir off`** *volume_name* is the name of the volume for which you want to enable client snapshot access.

Restoring files from snapshots

Why you restore files

You might need to restore a file from a snapshot if the file was accidentally erased or corrupted.

Note

If you have purchased the SnapRestore option, you can automatically restore files or volumes from snapshots with one command. To restore a volume, see "Deciding when to use SnapRestore" on page 57. To restore a single file, see "Reverting a file to a selected snapshot" on page 65.

How to restore a file

To restore a file from a snapshot, complete the following steps.

Step	Action
1	If the original file still exists and you do not want it overwritten by the snapshot file, from your UNIX or Windows client, rename the original file or move it to a different directory.
2	Locate the snapshot containing the version of the file you want to restore.
3	Copy the file from the .snapshot directory to the directory in which the file originally existed.

Creating snapshot schedules

About creating snapshot schedules

Data ONTAP provides a default snapshot schedule for each volume. You can configure the schedule to fit your needs. This schedule creates snapshots automatically and deletes old snapshots after a predetermined amount of time.

The default snapshot schedule

When you install Data ONTAP on a filer, it creates a default snapshot schedule. The default snapshot schedule automatically creates one nightly snapshot Monday through Saturday at midnight, and four hourly snapshots at 8 a.m., noon, 4 p.m., and 8 p.m. Data ONTAP retains the two most recent nightly snapshots and the six most recent hourly snapshots, and deletes the oldest nightly and hourly snapshots when new snapshots are created. You can see an example of the output for the default schedule in "Default snap schedule command results" on page 31.

User-specified snapshot schedules

There are three types of schedules that you can set up to run automatically using the `snap sched` command. The following table describes the three types.

Type	Description
Weekly	Data ONTAP creates these snapshots every Sunday at midnight. Weekly snapshots are called weekly.n, where *n* is an integer. The most recent weekly snapshot is weekly.0, and weekly.1 is the next most recent weekly snapshot.
Nightly	Data ONTAP creates these snapshots every night at midnight, except when a weekly snapshot is scheduled to occur at the same time. Nightly snapshots are called nightly.n, where *n* is an integer. The most recent nightly snapshot is nightly.0, and nightly.1 is the next most recent nightly snapshot.

Type	Description
Hourly	Data ONTAP creates these snapshots on the hour or at specified hours, except at midnight if a nightly or weekly snapshot is scheduled to occur at the same time. Hourly snapshots are called hourly.n, where *n* is an integer. The most recent hourly snapshot is hourly.0, and hourly.1 is the next most recent hourly snapshot.

Note

When Data ONTAP creates a weekly, nightly, or hourly snapshot, the value of *n* is adjusted for all the weekly, nightly, or hourly snapshots. The higher the value of *n*, the older the snapshot.

Displaying the snapshot schedule

To display the snapshot schedule for a volume on a filer, complete the following step.

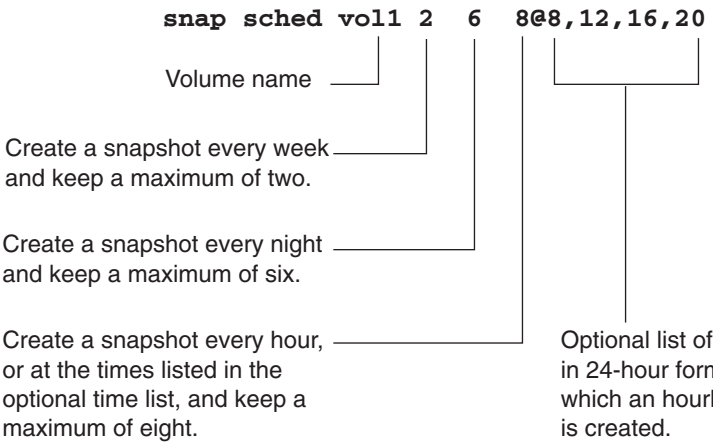
Step	Action
1	Enter the following command: **`snap sched [volume_name]`** **Note** If you do not specify a volume name, `snap sched` displays the snapshot schedule for each volume on the filer.

Example: The following is an example of the `snap sched` command output:

```
filerA>snap sched vol1
Volume vol1: 2 6 8@8,12,16,20
```

What the snap sched command arguments mean

The following illustration explains the arguments in a sample `snap sched` command output.



Snapshot schedule results: This schedule keeps the two most recent weekly snapshots, the six most recent nightly snapshots, and the eight most recent hourly snapshots, created at 8 a.m., noon, 4 p.m., and 8 p.m. every day. Whenever the snapshot schedule creates a new snapshot of a particular type, it deletes the oldest one and renames the existing ones. On the hour, for example, the filer deletes hourly.7, renames hourly.0 to hourly.1, and so on.

Note

If you omit the `@` argument specifying the hours for the hourly snapshots, Data ONTAP creates a snapshot every hour. You cannot specify the times for the nightly and weekly snapshots.

Strategies for creating a snapshot schedule

Following are some strategies for scheduling and retaining snapshots:

- If users rarely lose files or typically notice lost files right away, use the default snapshot schedule. This schedule creates no weekly snapshot; it creates a snapshot every night and keeps two; and it creates hourly snapshots at 8 a.m., noon, 4 p.m., and 8 p.m, and keeps six. Following is the default snapshot command:

  `snap sched` *`volume_name`* `0 2 6@8,12,16,20`

- If users commonly lose files or do not typically notice lost files right away, Network Appliance recommends that you delete the snapshots less often than you would if you used the default schedule.

Following is the recommended snapshot schedule for this situation. It keeps two weekly snapshots, six nightly snapshots, and eight hourly snapshots:

```
snap sched vol1 2 6 8@8,12,16,20
```

On many systems, only 5% or 10% of the data changes each week, so the snapshot schedule of six nightly and two weekly snapshots consumes 10% to 20% of disk space. Considering the benefits of snapshots, it is worthwhile to reserve this amount of disk space for snapshots. For more information on how snapshots consume disk space, see "Understanding snapshot disk consumption" on page 37.

- You can create different snapshot schedules for different volumes on a filer. On a very active volume, schedule snapshots every hour and keep them for just a few hours, or turn off snapshots. For example, the following schedule creates a snapshot every hour and keeps the last three:

  ```
  snap sched vol2 0 0 3
  ```

  This schedule does not consume much disk space, and it lets users recover files in recent snapshots as long as they notice their mistake within a couple of hours.

- When you create a new volume on a filer, the new volume inherits the snapshot schedule from the root volume. After you use the volume for a while, check how much disk space the snapshots consume and how often users need to recover lost files, and then adjust the schedule as necessary.

Changing the snapshot schedule

To change the automatic snapshot schedule for a specific volume, complete the following step.

Step	Action
1	Enter the following command: **`snap sched volume_name weekly nightly hourly@n,n,....`** *volume_name* is the name of the specific volume for the snapshot. *weekly* is the number of weekly snapshots to keep. *nightly* is the number of nightly snapshots to keep. *hourly* is the number of hourly snapshots to keep. *n,n,...* specifies the hours at which to create the hourly snapshots. **Note** A zero in any of the three schedules (weekly, nightly, hourly) disables snapshots for that interval.

Default snap schedule command results

This is the default automatic snapshot schedule:

```
snap sched volx 0 2 6 @8,12,16,20
```

The following example lists the snapshots created using the default schedule (where January 11 is a Sunday):

```
ls -lu .snapshot
total 64
drwxrwsrwx  1 root 4096 Jan 14 12:00 hourly.0
drwxrwsrwx  1 root 4096 Jan 14 08:00 hourly.1
drwxrwsrwx  1 root 4096 Jan 13 20:00 hourly.2
drwxrwsrwx  1 root 4096 Jan 13 16:00 hourly.3
drwxrwsrwx  1 root 4096 Jan 13 12:00 hourly.4
drwxrwsrwx  1 root 4096 Jan 13 08:00 hourly.5
drwxrwsrwx  1 root 4096 Jan 14 00:00 nightly.0
drwxrwsrwx  1 root 4096 Jan 13 00:00 nightly.1
```

Note

Daily snapshots are created at midnight of each day except Sunday, and weekly snapshots are created at midnight on Sunday. Only one snapshot is created at a time. If a weekly snapshot is being created, for instance, no daily or hourly snapshot will be created even if one is scheduled.

Disabling and enabling automatic snapshots

You can disable automatic snapshots for a period of time without changing the automatic snapshot schedule.

To temporarily disable or enable automatic snapshots, complete the following step.

Step	Action	
1	**If you want to...**	**Then...**
	Disable automatic snapshots	Enter the following command: **`vol options volume_name nosnap on`** *volume_name* is the name of the volume for which you want to disable automatic snapshots.
	Enable automatic snapshots	Enter the following command: **`vol options volume_name nosnap off`** *volume_name* is the name of the volume for which you want to enable automatic snapshots.

Note

You can also disable automatic snapshots by changing the snapshot schedule so that no snapshots are scheduled. To do this, enter the following command:

`snap sched volume_name 0 0 0`

Creating snapshots

Automatic or manual snapshots

The `snap sched` command automatically creates snapshots at preset intervals in a snapshot schedule. You can also create snapshots manually at any time.

Creating snapshots manually

To create a snapshot manually, complete the following step.

Step	Action
1	Enter the following command: **`snap create volume_name snapshot_name`** *volume_name* is the name of the volume on which you want to create the snapshot. *snapshot_name* is the name you want to give the snapshot.

Note

The `snap create` command does not accept a snapshot name containing a slash (/); therefore, it is not possible to enter a specific path for the snapshot file.

Finding the snapshot you need

Why you need to access a particular snapshot

You might need to access a particular snapshot (a version of a file at a particular point in time) because a file was changed, corrupted, or erased and the problem was not noticed until after one or more snapshots of it were created. When you start looking for the *version* of the file you need, you will look for it by means of the *access time* of the snapshot.

File version versus file access time

The *version* of a file refers to the last time the file was modified before a snapshot was created. The *access time* of a file refers to the snapshot creation time for a file, whether or not any modifications were made to that file.

Finding all file versions in snapshots

The best way to find all versions of a particular file preserved in snapshots is to use the `ls` command from an NFS client, or to use the Find function in Windows.

To find all the versions of a particular file in the snapshots, complete the following step.

Step	Action	
1	**If you are using...**	**Then...**
	An NFS client	Enter the following command: **`ls -l filename .snapshot/*/file_name`** **Result:** A list is displayed of all versions of the requested file.
	A CIFS client	1. Choose Find from the Windows Start Menu and select Files and Folders. **Result:** A search window opens prompting you for a directory and file name. 2. In the search window, enter the file name to search for in the ~snapshot directory.

NFS client example: The following example shows how to find all versions of the myfile.txt file:

```
ls -l myfile.txt .snapshot/*/myfile.txt
-rw-r--r--  1 smith 0 Jan 14 09:40  myfile.txt
-rw-r--r--  1 smith 0 Jan 13 18:39  .snapshot/nightly.0/myfile.txt
-rw-r--r--  1 smith 0 Jan 12 19:17  .snapshot/nightly.1/myfile.txt
```

The version of myfile.txt in the active file system was last modified on January 14, but the old versions available in the snapshots were modified on January 13 and January 12. Although users can use standard UNIX commands to read the saved versions of myfile.txt, they cannot modify or delete these older versions because everything in the .snapshot directory is read-only.

CIFS client example: If a user maps the home share to drive F: and wants to find all versions of myfile.txt in snapshots, the user can choose Find from the Windows Start menu to search for myfile.txt in the f:\~snapshot folder.

How to determine access times from an NFS client

When Data ONTAP creates a snapshot, the access time of each file in the snapshot is updated to the snapshot creation time.

To determine when snapshots were created, complete the following step.

Step	Action
1	From an NFS client, enter the following command: **`ls -lu filename .snapshot/*/file_name`**

Example: Following is an example of the `ls -lu` command:

```
ls -lu myfile.txt .snapshot/*/myfile.txt
-rw-r--r--  1 smith 0 Jan 14 09:40  myfile.txt
-rw-r--r--  1 smith 0 Jan 14 00:00  .snapshot/nightly.0/myfile.txt
-rw-r--r--  1 smith 0 Jan 13 00:00  .snapshot/nightly.1/myfile.txt
```

Note

On a UNIX client, if you use `ls -l` instead of `ls -lu` to list the snapshot creation times, the times are not necessarily all different. The times listed by `ls -l` reflect the modification times of the directory at the time of each snapshot, and are not related to the times at which the snapshots are created.

How to determine access times from a CIFS client

You can determine the access time of a file from a CIFS client by checking its properties.

Understanding snapshot disk consumption

Importance of understanding snapshot disk consumption

You need to understand the amount of disk space that snapshots consume and are likely to consume to determine

- How often to create a snapshot
- How long to keep a snapshot
- How much disk space you need for the snapshot reserve

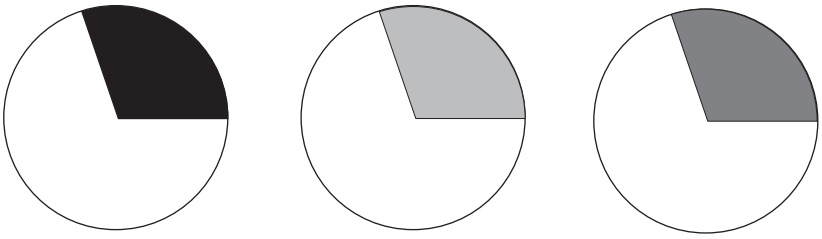
How snapshots consume disk space

Data ONTAP preserves pointers to all the disk blocks currently in use at the time the snapshot is created. When a file is changed, the snapshot still points to the disk blocks where the file existed before it was modified, and changes are written to new disk blocks.

Disk consumption is minimized by preserving individual disk blocks rather than whole files. Snapshots begin to consume extra space only when files in the active file system are changed or deleted. When this happens, the original file blocks are still preserved as part of one or more snapshots, but in the active file system the changed blocks are rewritten to different locations on disk or removed as active file blocks entirely. The result is that, in addition to the disk space used by blocks in the modified active file system, disk space used by the original blocks is still reserved in snapshots to reflect what the active file system was before the change.

Example of how snapshots consume disk space

The following illustration shows how snapshots consume disk space before and after you delete a file named myfile.txt.



Before any snapshot is taken, disk space is consumed by the active file system only.

After a snapshot is taken, the active file system and snapshot point to the same disk blocks. The snapshot does not use extra disk space.

After *myfile.txt* is deleted from the active file system, the snapshot still includes the file and references its disk blocks. That's why deleting active file system data does not always free disk space.

Space used by the active file system

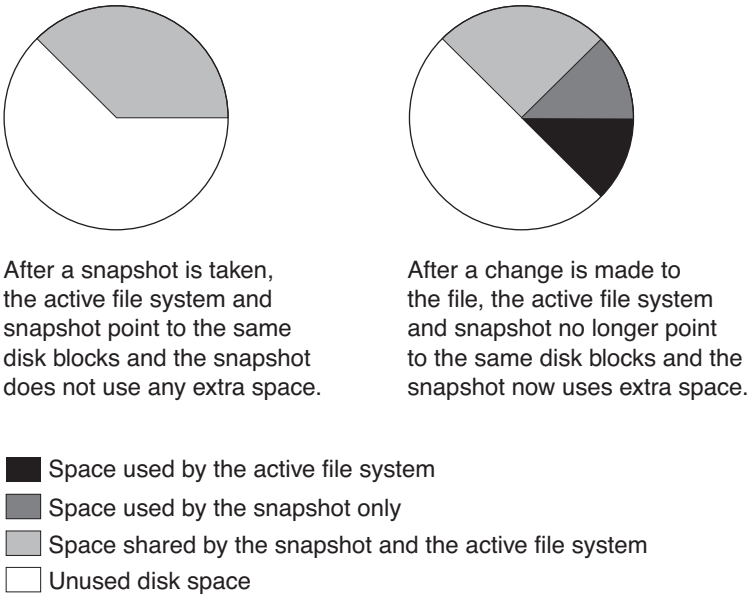
Space used by the snapshot only

Space shared by the snapshot and the active file system

Unused disk space

Example of how changing file content consumes disk space

Changing the contents of the myfile.txt file creates a situation similar to the one illustrated above. New data written to myfile.txt cannot be stored in the same disk blocks as the current contents because the snapshot is using those disk blocks to store the old version of myfile.txt. Instead, the new data is written to new disk blocks. As the following illustration shows, there are now two separate copies of myfile.txt on disk—a new copy in the active file system and an old one in the snapshot.



Disk consumption by multiple snapshots

Multiple snapshots can reference the same file without consuming extra disk space. Suppose a snapshot contains a 1-MB file that has not changed since Data ONTAP created the snapshot. If you delete that file from the active file system, the snapshot then consumes 1 MB of disk space.

The same version of that 1-MB file might be referenced by several snapshots: hourly.0, hourly.1, and hourly.2. If those snapshots all contain the 1-MB file that has not changed since Data ONTAP created them, only 1 MB of disk space is consumed by the snapshots even though all three snapshots contain the file. This is because they are all pointing to the same disk blocks.

For detailed information

The following sections discuss how to reserve disk space for snapshots and how to monitor snapshot disk space usage:

- "Monitoring snapshot disk consumption" on page 40
- "Displaying snapshot disk consumption statistics" on page 41
- "Understanding the snapshot reserve" on page 45
- "Changing the snapshot reserve" on page 48

Monitoring snapshot disk consumption

Displaying disk consumption

The `df` command displays the amount of free space on a disk. It treats snapshots as a partition different from the active file system.

To display information about snapshot disk consumption, complete the following step.

Step	Action
1	From the filer console or telnet session, enter the following command: **`df`**

Example: Following is a partial sample `df` command output:

```
filerA> df

Filesystem            kbytes    used    avail   capacity
/vol/vol0             3000000  2000000  1000000  65%
/vol/vol0/.snapshot   1000000  500000   500000   50%
```

Explanation of sample output: In the sample output, the kbytes column shows that the vol0 volume contains 3,000,000 KB (3 GB) of disk space for the active file system and 1,000,000 KB (1 GB) of disk space reserved for snapshots, for a total for 4,000,000 KB (4 GB) of disk space. In this example, 66% of the active disk space is used and 33% is available. Note that the capacity percentage rounds down to 65%. The 1,000,000 KB (1 GB) of disk space for snapshots represents 25% of the volume capacity, of which 500,000 KB (0.5 GB) is used and 500,000 KB (0.5 GB) is available, so that the space for snapshots is at 50% capacity.

Note that the 50% figure is not 50% of disk space, but 50% of allotted snapshot space. If allotted snapshot space is exceeded, this number will be over 100%.

It is important to understand that the /vol/vol0/.snapshot line counts data that exists only in a snapshot. The snapshot calculation does not include snapshot data that is shared with the active file system.

Displaying snapshot disk consumption statistics

How to display disk consumption for snapshots in a volume

The `snap list` command shows the amount of disk space consumed by snapshots in a specified volume. This command enables you to see how much disk space each snapshot uses and helps you determine an appropriate snapshot reserve. The `snap list` command also shows whether a snapshot is currently being used for backup and whether it is needed for a restartable dump or another backup application.

Listing snapshot statistics for a volume

To display the snapshot statistics for a volume, complete the following step.

Step	Action
1	Enter the following command line: **`snap list volume_name`** *volume_name* is the name of the volume for which you want statistics.

Sample snap list command output

Following is an example of the `snap list` command output. If you do not specify a volume name in the command, the output contains statistics about each volume in the system.

```
filerA> snap list vol0
Volume vol0
%/used       %/total    date          name
----------  ----------  ------------  --------
0% ( 0%)    0% ( 0%)    Jan 19 08:01  hourly.0
1% ( 1%)    1% ( 1%)    Jan 19 00:01  nightly.0
2% ( 2%)    2% ( 2%)    Jan 18 20:01  hourly.1
3% ( 2%)    2% ( 2%)    Jan 18 16:01  hourly.2
3% ( 2%)    3% ( 2%)    Jan 18 12:01  hourly.3
5% ( 3%)    4% ( 3%)    Jan 18 00:01  nightly.1
7% ( 4%)    6% ( 4%)    Jan 17 00:00  nightly.2
8% ( 4%)    7% ( 4%)    Jan 16 00:01  nightly.3
10%( 5%)    9% ( 4%)    Jan 15 00:01  nightly.4
```

How the snap list output is calculated

The %/used column: The %/used column shows space consumed by snapshots as a percentage of disk space being used in the volume. The first number is cumulative for all snapshots listed so far, and the second number is for the specified snapshot alone.

- The first number is equal to

$$100\% \times \frac{\text{cumulative snapshot space}}{\text{cumulative snapshot space + file system space}}$$

- The second number is equal to

$$100\% \times \frac{\text{this snapshot}}{\text{this snapshot + file system space}}$$

The %/total column: The %/total column shows space consumed by snapshots as a percentage of total disk space (both space used and space available) in the volume.

- The first number is equal to

$$100\% \times \frac{\text{cumulative snapshot space}}{\text{total disk space in this volume}}$$

  Cumulative snapshot space is the total space used by this snapshot and all other more recent snapshots (the ones preceding this snapshot in the `snap list` output).

- The second number is equal to

$$100\% \times \frac{\text{this snapshot}}{\text{total disk space in this volume}}$$

Summary of the snap list output

The %/used number is the most useful for planning the snapshot reserve because it is more likely to remain constant as the file system fills.

The information in "Sample snap list command output" on page 41 shows a volume that keeps five nightly snapshots and four hourly snapshots.

The sample output shows that the overhead for snapshots is only 10%, so the default snapshot reserve of 20% seems to be a waste of disk space. Assuming that this pattern of change holds, a reserve of 12% to 15% provides a safe margin to

ensure that deleting files frees disk space when the active file system is full. For information about the snapshot reserve, see “Understanding the snapshot reserve” on page 45.

The values in parentheses, which show the space used by an individual snapshot, are useful in identifying a particular snapshot to delete when the file system is full. However, deleting a particular snapshot does not necessarily release the amount of disk space indicated, because other snapshots might be referring to the same blocks.

About using cumulative snapshot values

If you do not want the amount of disk space consumed by all snapshots to exceed a certain percentage of the used disk space, use the cumulative values in the `snap list` output to determine which snapshots to delete. In the preceding example, if you do not want more than 5% of used disk space to be spent by snapshots, delete all snapshots listed below nightly.1 in the `snap list` output; that is, nightly.2, nightly.3, and nightly.4. After deleting the snapshots, nightly.1 and all the other more recent snapshots consume 5% of the used disk space.

Displaying snapshot use and dependencies

The `snap list` command displays the notation “busy” after the name of a snapshot if the snapshot is being actively used by an application, as shown the example below for the snapshot hourly.0. Busy snapshots cannot be deleted.

```
snap list vol0

Volume vol0
 %/used    %/total    date          name
--------  ------     ------------  --------
0% ( 0%)  0% ( 0%)   Jan 19 08:01  hourly.0 (busy)
1% ( 1%)  1% ( 1%)   Jan 19 00:01  nightly.0(snapmirror)
2% ( 2%)  2% ( 2%)   Jan 18 20:01  hourly.1 (busy,snapmirror)
3% ( 2%)  2% ( 2%)   Jan 18 16:01  hourly.2
3% ( 2%)  3% ( 2%)   Jan 18 12:01  hourly.3
5% ( 3%)  4% ( 3%)   Jan 18 00:01  nightly.1 (backup[9])
7% ( 4%)  6% ( 4%)   Jan 17 00:00  nightly.2
8% ( 4%)  7% ( 4%)   Jan 16 00:01  nightly.3
10%( 5%)  9% ( 4%)   Jan 15 00:01  nightly.4
```

The `snap list` command also displays the name of an application next to a snapshot name if the application needs the snapshot currently or at a later time. For example, “backup” is displayed next to the snapshot name to show that the snapshot is the result of a `dump` command transfer that was interrupted but is restartable. The number following “backup” is the backup ID assigned by the

`backup status` command. The notation “snapmirror” next to the snapshot name means that SnapMirror is retaining the snapshot to maintain a source-destination relationship.

snap list performance after a snap restore file operation

If you restore files with the `snap restore` command, and then issue the `snap list` command, the `snap list` command might require from several seconds to several minutes to complete. This condition persists until the snapshot version from which your restored the file is purged from the system after reaching the end of its normal snapshot retention cycle.

For more information see “Reverting a file to a selected snapshot” on page 65.

Understanding the snapshot reserve

What the snapshot reserve is

The snapshot reserve specifies a set percentage of disk space for snapshots. By default, the snapshot reserve is 20% of disk space. The snapshot reserve can be used only by snapshots, not by the active file system. This means that if the active file system runs out of disk space, any disk space still remaining in the snapshot reserve is not available for active file system use.

Note

Although the active file system cannot consume disk space reserved for snapshots, snapshots can exceed the snapshot reserve and consume disk space normally available to the active file system.

Snapshot reserve management tasks

Snapshot reserve management involves the following tasks:

- Ensuring that enough disk space is set aside for snapshots so that they do not consume active file system space
- Keeping disk space consumed by snapshots below the snapshot reserve
- Ensuring that the snapshot reserve is not so large that it wastes space that could be used by the active file system

Use of deleted active file disk space

When enough disk space is available for snapshots in the snapshot reserve, deleting files in the active file system frees disk space for new files, while the snapshots that reference those files consume only the space in the snapshot reserve. If Data ONTAP created a snapshot when the disks were full, deleting files from the active file system would not create any free space because everything in the active file system would also be referenced by the newly created snapshot. Data ONTAP would have to delete the snapshot before it could create any new files.

Example: The following example shows how disk space being freed by deleting files in the active file system ends up in the snapshot:

If Data ONTAP creates a snapshot when the active file system is full and there is still space remaining in the snapshot reserve, the output from the `df` command—which displays statistics about the amount of disk space on a volume—is as follows:

```
Filesystem            kbytes    used     avail     capacity
/vol/vol0/            3000000   3000000  0         100%
/vol/vol0/.snapshot   1000000   500000   500000    0%
```

If you delete 100,000 KB (0.1 GB) of files, the disk space used by these files is no longer part of the active file system, so the space is reassigned to the snapshots instead.

Explanation: Data ONTAP reassigns 100,000 KB (0.1 GB) of space from the active file system to the snapshot reserve. Because there was reserve space for snapshots, deleting files from the active file system freed space for new files. If you enter the `df` command again, the output is as follows:

```
Filesystem            kbytes    used     avail     capacity
/vol/vol0/            3000000   2900000  100000    97%
/vol/vol0/.snapshot   1000000   600000   400000    60%
```

Snapshots can exceed reserve

There is no way to prevent snapshots from consuming disk space greater than the amount reserved for them; therefore, it is important to reserve enough disk space for snapshots so that the active file system always has space available to create new files or modify existing ones.

Example: Consider what would happen in the example if all files in the active file system were deleted. Before the deletion, the `df` output was as follows:

```
Filesystem            kbytes    used     avail     capacity
/vol/vol0/            3000000   3000000  0         100%
/vol/vol0/.snapshot   1000000   500000   500000    50%
```

After the deletion, the `df` command generates the following output:

```
Filesystem            kbytes    used     avail     capacity
/vol/vol0/            3000000   2500000  500000    83%
/vol/vol0/.snapshot   1000000   3500000  0         350%
```

Explanation: The entire 3,000,000 KB (3 GB) in the active file system is still being used by snapshots, along with the 500,000 KB (0.5 GB) that were being used by snapshots before, making a total of 3,500,000 KB (3.5 GB) of snapshot data. This is 2,500,000 KB (2.5 GB) more than the space reserved for snapshots; therefore, 2.5 GB of space that would be available to the active file system is now unavailable to it. The post-deletion output of the `df` command lists this unavailable space as “used” even though no files are stored in the active file system.

Recovering disk space for file system use

Whenever snapshots consume more than 100% of the snapshot reserve, the system is in danger of becoming full. In this case, you can create files only after you delete enough snapshots.

Example: If 500,000 KB (0.5 GB) of data is added to the active file system in the preceding example, a `df` command generates the following output:

```
Filesystem            kbytes    used     avail     capacity
/vol/vol0             3000000   3000000  0         100%
/vol/vol0/.snapshot   1000000   3500000  0         350%
```

Explanation: As soon as Data ONTAP creates a new snapshot, every disk block in the file system is referenced by some snapshot. Therefore, no matter how many files you delete from the active file system, there is still no room to add any more. The only way to recover from this situation is to delete enough snapshots to free more disk space.

See "Displaying snapshot disk consumption statistics" on page 41 for information about how to use the `snap list` command to determine which snapshots to delete.

Changing the snapshot reserve

Changing the snapshot reserve

After you understand how snapshots consume disk space and how the snapshot reserve works, you can change the snapshot reserve when needed.

To change the percentage of disk space used for the snapshot reserve, complete the following step.

Step	Action
1	Enter the following command: **`snap reserve volume_name percent`** *volume_name* is the name of the volume on the filer. *percent* is the percentage of disk space you want to reserve for snapshots.

Example: `snap reserve vol0 25`

Saving disk space by using file folding

What file folding means and how it saves disk space

File folding describes the process of checking the data in the most recent snapshot, and, if it is identical to the snapshot currently being created, just referencing the previous snapshot instead of taking up disk space writing the same data in the new snapshot. Disk space is saved by sharing unchanged file blocks between the active version of the file and the version of the file in the latest snapshot, if any.

The filer must compare block contents when folding a file, so there is a tradeoff between performance and space utilization.

If the folding process reaches a maximum limit on memory usage, it is suspended. When memory usage falls below the limit, the processes that were halted are restarted.

How to turn file folding on

To turn file folding on, complete the following step.

Step	Action
1	Enter the following command: `options cifs.snapshot_file_folding.enable on`

How to turn file folding off

To turn file folding off, complete the following step.

Step	Action
1	Enter the following command: `options cifs.snapshot_file_folding.enable off` This option is off by default.

Note

This option is not available to NFS users in this release.

Deleting snapshots

How to determine which snapshots to delete on the basis of size

You can use the `snap list` command output to determine which snapshots to delete to free the most disk space.

In "Sample snap list command output" on page 41, the cumulative disk space used by snapshots gradually increases from top to bottom.

For example, in the %/used column, the cumulative space used by hourly.1 is 2% and the cumulative space used by hourly.2 is 3%. This is not always the case.

Before trying to conserve space by deleting a large snapshot file, examine the cumulative values in the `snap list` output. If two adjacent snapshot files show little difference in their cumulative values, most of the data referenced by these snapshots is the same. In this case, deleting only one of the snapshots does not free much disk space.

In many cases, you can use the default snapshot schedule and the default snapshot reserve because these settings are appropriate for most environments. When you create a new volume on a filer, the new volume inherits the snapshot schedule from the root volume. After you use the volume for several days, check how much disk space the snapshots are consuming in the volume. If the amount seems high, decrease the amount of time that snapshots are kept or increase the snapshot reserve.

As you use snapshots, continue to watch the statistics change over time. The statistics help you gain a better understanding of how snapshots use disk space.

Caution

As a general rule, avoid deleting snapshots that are not the product of the `snap sched` command (for example, snapshots generated by SnapMirror or SnapVault commands). Deleting these snapshots could halt the SnapMirror or SnapVault processes. An exception would be snapshots left over from old SnapMirror relationships that you no longer want to maintain. See "Releasing partners from a SnapMirror relationship" on page 142.

Deleting a snapshot manually

The `snap sched` command deletes obsolete regularly scheduled snapshots automatically; however, you might want to delete a snapshot before the preset interval to increase disk space or because it is a manual snapshot that is no longer needed but is not going to be automatically deleted.

To delete a snapshot manually from a specific volume, complete the following step.

Step	Action
1	Enter the following command: **`snap delete volume_name snapshot_name`** *volume_name* is the name of the volume that contains the snapshot to delete. *snapshot_name* is the specific snapshot to delete. **Note** To delete all snapshots on a volume, use the `-a` parameter: **`snap delete -a`** *volume_name*

Deleting a snapshot manually if the snapshot is locked

If a snapshot is locked, the `snap delete` operation fails until you execute a `snapmirror release` (see "How to release a source from a SnapMirror relationship" on page 142) or `snapvault release` (see "Releasing SnapVault relationships" on page 240) command to unlock the snapshot. Snapshots are locked because SnapMirror or SnapVault is maintaining them for the next update. Deleting a locked snapshot would prevent SnapMirror or SnapVault from correctly replicating a file or volume as specified in the schedule you set up.

Example of how to delete a locked SnapMirror snapshot

The following example shows how you would delete a SnapMirror snapshot that was locked because SnapMirror needed it for an update.

```
filerA> snap delete vol0 oldsnap
Can't delete oldsnap: snapshot is in use by snapmirror.
Use 'snapmirror destinations -s' to find out why.
filerA> snapmirror destinations -s vol0
Path        Destination
/vol/vol0   filerB:vol0
filerA> snapmirror release vol0 filerB:vol0
filerA> snap delete vol0 oldsnap
```

Example of how to delete a locked SnapVault snapshot

The following example shows how you would delete a SnapVault snapshot that was locked because SnapVault needed it for an update.

```
filerA> snap delete vol0 oldsnap
Can't delete oldsnap: snapshot is in use by snapvault.
Use 'snapvault status -l' to find out why.

filerA> snapvault status -l
SnapVault client is ON.
Source:                     filerA:/vol/vol0/qt3
Destination                 filerB:/vol/sv_vol/qt3
....

filerA> snapvault release /vol/vol0/qt3 filerB:/vol/sv_vol/qt3

filerA> snap delete vol0 oldsnap
```

Renaming snapshots

Why to rename a snapshot

You might want to rename a `snap sched`-generated snapshot if it contains data that you want to save. You can prevent it from being overwritten or deleted by the `snap sched` command process, which deletes regularly scheduled snapshots automatically, by using the `snap rename` command.

How to rename a snapshot

To rename a snapshot, complete the following step.

Step	Action
1	Enter the following command: **`snap rename volume_name from_name to_name`** *volume_name* is the name of the volume that contains the snapshot to rename. *from_name* is the current name of the snapshot to rename. *to_name* is the new name you want to give to the snapshot.

Example: `snap rename vol0 hourly.2 MyDataSave`

Data Reversion Using SnapRestore 3

About this chapter

This chapter describes how to restore data to a file or volume using the Data ONTAP SnapRestore feature. Read this chapter if you want to recover data that is no longer available or if you are testing a volume or file and want to restore that volume or file to pre-test conditions.

Note

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

Topics in this chapter

This chapter discusses the following topics:

- "Understanding SnapRestore" on page 56
- "Deciding when to use SnapRestore" on page 57
- "Installing the SnapRestore license" on page 60
- "Reverting a volume to a selected snapshot" on page 61
- "Reverting a file to a selected snapshot" on page 65
- "Obtaining correct incremental backups after reversion" on page 70

Understanding SnapRestore

What SnapRestore does

SnapRestore enables you to revert a local volume or file on a filer quickly to the state it was in when a particular snapshot was taken. In most cases, reverting a file or volume is much faster than restoring files from tape or copying files from a snapshot to the active file system.

How SnapRestore works

After you select a snapshot for reversion, the filer reverts the specified file or the volume to the data and timestamps that it contained when the selected snapshot was taken. Data that was written after the selected snapshot was taken is lost. If the volume you select for reversion is a root volume, the filer reboots the system.

What SnapRestore reverts

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

When to use SnapRestore

You use SnapRestore to recover from data corruption. If a primary storage system application corrupts data files in a volume, you can revert the volume or specified files in the volume to a snapshot taken before the data corruption.

Why use SnapRestore rather than copying from a snapshot

SnapRestore carries out snapshot restoration more quickly, using less disk space, than an administrator can achieve by manually copying volumes, qtrees, directories, or large files to be restored from the filer's snapshot system to the active file system. A large volume directory restore can be carried out in a few seconds using the SnapRestore feature.

SnapRestore can restore large volumes or files even if space limitations would prevent restoring by copying from a snapshot.

Deciding when to use SnapRestore

Considerations before using SnapRestore

You must take into account the following considerations before deciding whether to use SnapRestore to revert a file or volume.

- If the volume that you need to restore is a root volume, it is easier to copy the files from a snapshot or restore the files from tape than to use SnapRestore because you can avoid rebooting. If you need to restore a corrupted file on a root volume, however, a reboot is not necessary.
- If you revert the whole root volume, the filer reboots with configuration files that were in effect when the snapshot was taken.
- If the amount of data to be recovered is large, SnapRestore is the preferred method, because it takes a long time to copy large amounts of data from a snapshot or to restore from tape.
- If a file to be recovered needs more space than the amount of free space in the active file system, you cannot restore the file by copying from the snapshot to the active file system. For example, if a 10-GB file is corrupted and only 5 GB of free space exists in the active file system, you cannot copy the file from a snapshot to recover the file. However, SnapRestore can quickly recover the file in these conditions. You do not have to spend time making the additional space available in the active file system.

Caution

SnapRestore lets you revert to a snapshot from a previous release of Data ONTAP. However, doing so can cause problems because of potential version incompatibilities and can prevent the filer from booting completely.

Prerequisites

You must meet these prerequisites before using SnapRestore:

- SnapRestore must be licensed on your filer.
- There must be at least one snapshot on your filer that you can select for the reversion.
- The volume to be reverted must be online.
- The volume to be reverted must not be a mirror used for data replication.

General cautions

Be sure that you understand the following facts before using SnapRestore:

- SnapRestore overwrites all data in the file or volume. After you use SnapRestore to revert to a selected snapshot, you cannot undo the reversion.
- If you revert to a snapshot taken before a SnapMirror snapshot, Data ONTAP can no longer perform an incremental update of the mirror; it must re-create the baseline version of the mirror.
- Snapshot deletions are irrevocable. If you delete a snapshot, you cannot recover the snapshot by using SnapRestore.
- After you revert a volume to a selected snapshot, you lose all the snapshots that were taken after the selected snapshot.
- Between the time you enter the `snap restore` command and the time when reversion is completed, Data ONTAP stops deleting and creating snapshots.
- If you are reverting a file from a snapshot, you can delete other snapshots, except for the snapshot you are reverting from.

Caution about reverting the root volume

Because the /etc directory of the root volume contains configuration information about the filer, reverting the root volume might change the filer configuration. In addition, restoring the root volume restores options settings for the entire filer to the settings that were in effect when the snapshot was taken. Reverting a root volume requires rebooting the system.

Preserving configuration files

To avoid reverting the configuration files, complete the following step.

Step	Action
1	Store all data that might need to be reverted in a volume other than the root volume. This ensures that you never need to revert the root volume.

Reverting a root volume

If you need to revert a root volume, complete the following step.

Step	Action
1	If the data you want to revert resides in the root volume, back up the /etc directory to another volume or another filer before using SnapRestore. After you revert the root volume, restore the /etc directory and reboot the filer. If you back up the /etc directory to another volume, you can use the `vol options` *volume* `root` command to make the filer reboot with that volume as the root volume. In this way, when the filer reboots during the reversion, it can use the correct settings in the /etc directory.

Installing the SnapRestore license

How to install the SnapRestore license code

To install the SnapRestore license code, complete the following step.

Step	Action
1	On the server, enter the following command: **`license add xxxxxxx`** *xxxxxxx* is the license code you purchased. This setting persists across reboots.

Reverting a volume to a selected snapshot

How to revert a volume

To revert a volume, complete the following steps.

Note

To cancel volume reversion, press Ctrl-C at any time before you press `y` in Step 8.

Step	Action	
1	Notify network users that you are going to revert a volume so they know that the current data in the volume will be replaced by that of the selected snapshot. **Note** NFS users should unmount the files and directories in the volume before the reversion. If they do not unmount the files and directories, they might get a “stale file handle” error message after the volume reversion.	
2	**If...**	**Then...**
	You know the name of the snapshot for reverting each volume you want to revert	Go to Step 6.
	You want to choose a snapshot from the list of snapshots available for reversion	Go to Step 3.

Step	Action
3	Enter the following command: **`snap restore[-f] [-t vol]` *`volume_name`*** The `-f` option suppresses the warning message and the prompt for confirmation. The `-t vol` option specifies the volume name to revert. *volume_name* is the name of the volume to be reverted. Enter the name only, not the complete path. You can enter only one volume name. **Result:** If you have not used the `-f` option, Data ONTAP displays a warning message and prompts you to confirm your decision to revert the volume.
4	Press **`y`** to confirm that you want to revert the volume. **Result:** Data ONTAP displays a list of snapshots.
5	Enter the name of the snapshot for reverting the volume, then go to Step 8. **Result:** Data ONTAP displays the name of the volume to be reverted and the name of the snapshot to be used for the reversion.

Step	Action	
6	Enter the following command: **`snap restore [-f] [-t vol] volume_name [-s snapshot_name]`** `-f` suppresses the warning message and the prompt for confirmation. This option is useful for scripts. `-t vol` *volume_name* specifies the volume name (*volume_name*) to revert. `-s` *snapshot_name* specifies the name of the snapshot from which to revert the data. You can enter only one snapshot name. *volume_name* is the name of the volume to be reverted. Enter the name only, not the complete path. You can enter only one volume name. **Result:** If you have not used the `-f` option, Data ONTAP displays a warning message and prompts you to confirm your decision to revert the volume.	
7	Press **`y`** to confirm that you want to revert the volume. **Result:** Data ONTAP displays the name of the volume and the name of the snapshot for the reversion and, if you have not used the `-f` option, prompts you to decide whether to proceed with the reversion.	
8	**If...**	**Then...**
	You want to continue with the reversion	Press **`y`**. **Result:** The filer reverts the volume from the selected snapshot. If you are reverting the root volume, the filer reboots the system.
	You do not want to proceed with the reversion	Press **`n`** or **`Ctrl-C`**. **Result:** The volume is not reverted and you are returned to a filer prompt.

Example:
```
filer> snap restore -t vol /vol/vol1 -s nightly.0

filer> WARNING! This will restore a volume from a snapshot into the
active filesystem.  If the volume already exists in the active
filesystem, it will be overwritten with the contents from the
snapshot.
Are you sure you want to do this? y

You have selected file /vol/vol1, snapshot nightly.0
Proceed with restore? y
```

Result: Data ONTAP restores the volume called vol1 at /vol/vol1.

After a volume is reverted with SnapRestore, all user-visible information (data and attributes) for that volume in the active file system is identical to that contained in the snapshot.

Reverting a file to a selected snapshot

About snap restore file reversion

The use of `snap restore` to revert a single file to a selected snapshot is practical when the size of the file is so large that you cannot copy the previous file version from the snapshot to the active file system.

Some notes concerning snap restore file reversion:

- You cannot use SnapRestore for single file reversion on files with NT streams, or on directories.
- If you restore single files with the `snap restore` command, and then issue the `snap list` command, the `snap list` command might require from several seconds to several minutes to complete. This condition persists until the snapshot version from which your restored the file is purged from the system after reaching the end of its normal snapshot retention cycle.

How to revert a file

To revert a single file (rather than a volume), complete the following steps.

Note

To cancel file reversion, press Ctrl-C at any time before you press **`y`** in Step 10.

Step	Action
1	Notify network users that you are going to revert a file so that they know that the current data in the file will be replaced by that of the selected snapshot. **Note** NFS users who try to access a reverted file without first reopening it might get a “stale file handle” error message after the volume reversion.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>2</td><td>If...</td><td>Then...</td></tr>
<tr><td></td><td>You know the name of the snapshot for reverting the file you want to revert</td><td>Go to Step 6.</td></tr>
<tr><td></td><td>You want to choose a snapshot from the list of snapshots available for reversion</td><td>Go to Step 3.</td></tr>
<tr><td>3</td><td colspan="2">Enter the following command:
<tt>snap restore [-f] [-t file] [-r restore_as_new_path] path_and_file_name</tt>
options can be one or more of the following:
<tt>-f</tt> suppresses the warning message and the prompt for confirmation.
<tt>-t file</tt> specifies that you are entering the name of a file to revert.
<tt>-r</tt> restore_as_new_path restores the file to a location different from (but in the same volume as) the location in the snapshot. For example, if you specify /vol/vol0/vol3/myfile as the argument to <tt>-r</tt>, SnapRestore reverts the file called myfile to the location /vol/vol0/vol3 instead of to the path in vol3 indicated by path_and_file_name.
path_and_file_name is the complete path to the name of the file to be reverted. You can enter only one path name.
A file can be restored only to the volume where it was originally. The directory structure to which a file is to be restored must be the same as specified in the path. If this directory structure no longer exists, you must recreate it before restoring the file.
Result: Data ONTAP displays a warning message and prompts you to confirm your decision to revert the file.</td></tr>
<tr><td>4</td><td colspan="2">Press <tt>y</tt> to confirm that you want to revert the file.
Result: Data ONTAP displays a list of snapshots.</td></tr>
</table>

Step	Action
5	Enter the name of the snapshot for reverting the file, then go to Step 8. **Result:** Data ONTAP displays the name of the file to revert and the name of the snapshot to be used for the reversion.
6	Enter the following command: **`snap restore[-f] [-t file] -s snapshot_name [-r restore_as_path] path_and_file_name`** *options* can be one or more of the following: `-f` suppresses the warning message and the prompt for confirmation. `-t file` specifies that you are entering the name of a file to revert. `-s` *snapshot_name* specifies the name of the snapshot from which to revert the data. `-r` *restore_as_path* restores the file to a location different from the location in the snapshot. For example, if you specify /vol/vol0/vol3/myfile as the argument to `-r`, SnapRestore reverts the file called myfile to the location /vol/vol0/vol3 instead of to the file structure indicated by the path in *path_and_file_name.* *path_and_file_name* is the complete path to the name of the file to be reverted. You can enter only one path name. A file can be restored only to the volume where it was originally. The directory structure to which a file is to be restored must be the same as specified in the path. If this directory structure no longer exists, you must recreate it before restoring the file. Unless you enter `-r` and a path name, only the file at the end of the *path_and_file_name* is reverted. You can enter only one path name. **Result:** If you have not used the `-f` option, Data ONTAP displays a warning message and prompts you to confirm your decision to revert the file.

Step	Action	
7	Press **y** to confirm that you want to revert the file. **Result:** Data ONTAP displays the name of the file and the name of the snapshot for the reversion and, if you have not used the `-f` option, prompts you to decide whether to proceed with the reversion.	
8	**If...**	**Then...**
	You want to continue with the reversion	Press **y**. **Result:** The filer reverts the file from the selected snapshot.
	You do not want to proceed with the reversion	Press **n** or **Ctrl-C**. **Result:** The file is not reverted and you are returned to a filer prompt.

Example:
```
filer> snap restore -t file /vol/vol1/users/jim/myfile
-s nightly.0

filer> WARNING! This will restore a file from a snapshot into the
active filesystem.  If the file already exists in the active
filesystem, it will be overwritten with the contents from the
snapshot.
Are you sure you want to do this? y

You have selected file /vol/vol1/users/jim/myfile, snapshot
nightly.0
Proceed with restore? y
```

Result: Data ONTAP restores the file called myfile to the existing volume and directory structure /vol/vol1/users/jim.

Example:
```
filer> snap restore -t file -s nightly.0
-r /vol/vol2/archive/eng/myfile /vol/vol2/users/jim/myfile

filer> WARNING! This will restore a file from a snapshot into the
active filesystem.  If the file already exists in the active
filesystem, it will be overwritten with the contents from the
snapshot.
Are you sure you want to do this? y
```

```
You have selected file /vol/vol1/users/jim/myfile, snapshot
nightly.0
Proceed with restore? y
```

Result: Data ONTAP restores the file called myfile to a new location at /vol/vol2/archive/eng.

After a file has been reverted with SnapRestore, all user-visible information (data and file attributes) for that file in the active file system is identical to that contained in the snapshot.

Obtaining correct incremental backups after reversion

Reversion effects on backup and restore

All files in a reverted volume have timestamps that are the same as those when the snapshot was created. After a reversion, incremental backup and restore operations on the file or volume cannot rely on the timestamps to determine what data needs to be backed up or restored.

Ensuring correct incremental backups after reversion

To ensure correct incremental backups, complete the following steps.

Step	Action
1	Perform a base-level backup of the volume after you restore it.
2	If you need to restore data from tape, use only backups created after the volume was restored.

Data Protection Using SnapMirror 4

About this chapter

This chapter discusses how to use the optional SnapMirror feature of Data ONTAP to copy data from specified volumes or qtrees to other volumes or qtrees and how to access and use the online copied data.

Note

You must have a SnapMirror license to use SnapMirror.

Topics in this chapter

This chapter discusses the following topics:

- "SnapMirror overview," on page 72
- "SnapMirror commands and configuration files," on page 79
- "Considerations when planning and running SnapMirror," on page 83
- "Setting up a basic SnapMirror operation," on page 86
- "Specifying destination filers on the source filer," on page 89
- "Defining source and destination through snapmirror.conf," on page 92
- "Enabling SnapMirror," on page 100
- "Initializing a SnapMirror destination," on page 102
- "Updating a destination manually," on page 110
- "Listing SnapMirror snapshots," on page 113
- "Checking SnapMirror data transfer status," on page 116
- "Checking SnapMirror data transfer logs," on page 125
- "Aborting a SnapMirror transfer," on page 130
- "Stabilizing (quiescing) destinations before taking a snapshot," on page 132
- "Turning off or changing scheduled updates for volumes or qtrees," on page 135
- "Turning off SnapMirror updates for the filer," on page 137
- "Converting a destination to a writable volume or qtree," on page 139
- "Releasing partners from a SnapMirror relationship," on page 142
- "Resynchronizing SnapMirror," on page 144
- "Copying from one destination to another in a series (cascading)," on page 151
- "SnapMirroring a volume to local tape," on page 156
- "How SnapMirror works with the dump command," on page 167
- "Fixing changes to SnapMirror elements," on page 168

SnapMirror overview

What SnapMirror is

The Data ONTAP SnapMirror feature enables an administrator to mirror snapshot images from a source volume or qtree to a partner destination volume or qtree, thus replicating source object data on destination objects at regular intervals. You can access the information on the destination volume or qtree to

- Provide users quick access to mirrored data in the event of a disaster that makes the source volume or qtree unavailable.
- Update the source to recover from disaster, data corruption (qtrees only), or user error
- Archive the data to tape
- Balance resource loads
- Back up or distribute the data to remote sites

Note

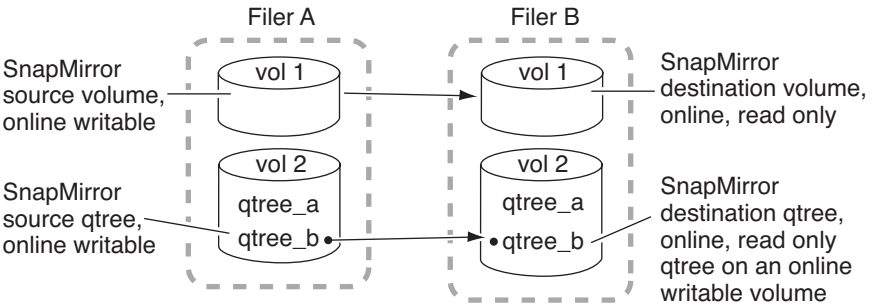
This chapter describes how to set up and use SnapMirror on filers configured for network attached storage (NAS). For a description of SnapMirror implementation and management on filers configured for a storage area network (SAN) environment, see the *SAN Setup Overview FCP 1.1 Guide*.

The components of SnapMirror

The basic SnapMirror deployment consists of the following components.

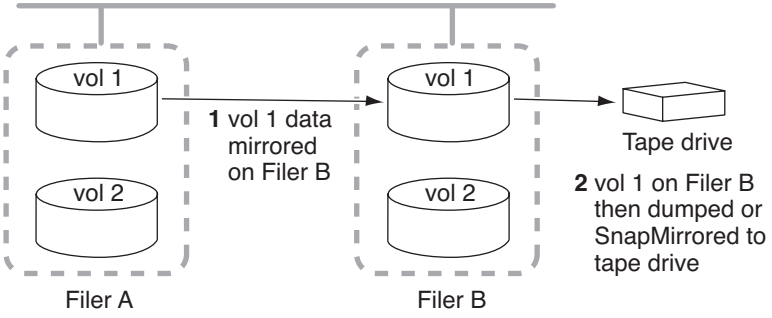
Source volumes or qtrees: SnapMirror source volumes and qtrees are writable data objects whose data is to be replicated. The source volumes and qtrees are the objects normally visible, accessible, and writable by the filer clients.

Destination volumes or qtrees: The SnapMirror destination volumes and qtrees are read-only objects, usually on a separate filer, to which the source volumes and qtrees are replicated. The destination volumes and qtrees are normally accessed by users only when a disaster takes down the source volumes or qtrees and the administrator uses SnapMirror commands to make the replicated data at the destination accessible and writable.



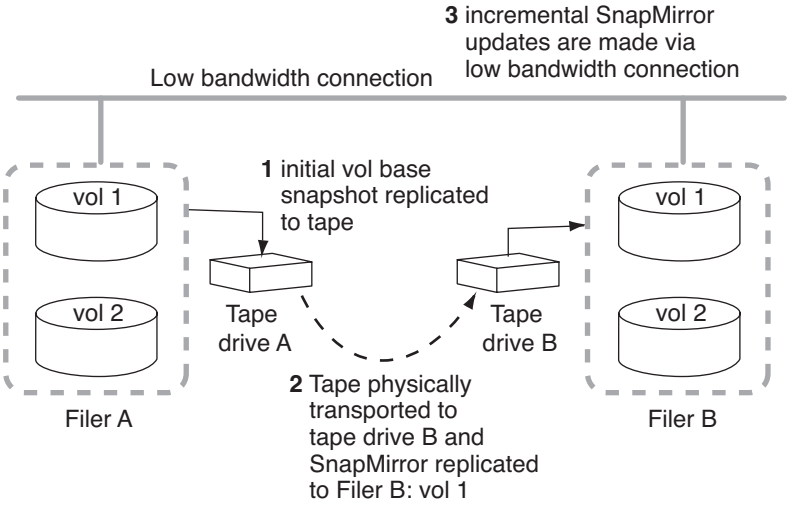
Source to destination to tape variation

A common variation to the basic SnapMirror backup deployment adds a tape backup of the destination volume. By running a tape backup off the SnapMirror destination volume, you do not subject the heavily user-accessed source volume to the performance degradation, system unavailability, and complexity of a direct tape backup.



Source to tape to destination variation

A SnapMirror deployment that supports SnapMirror replication over low-bandwidth connections accommodates an initial mirroring between a source and destination volume via physically transported tape. Once the large-sized base snapshot replication has been carried out, smaller-sized, incremental snapshot updates can be carried out over the low-bandwidth connection.

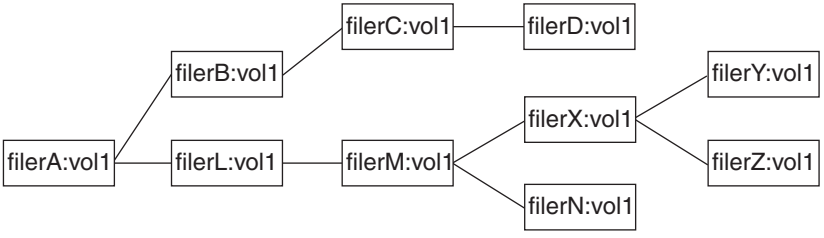


Cascading destinations variation

A variation on the basic SnapMirror deployment *and function* involves a writable source volume replicated to multiple read-only destinations. The function of this deployment is to make a uniform set of data available on a read-only basis to users from various locations throughout a network and to allow for updating that data uniformly at regular intervals.

Note

The cascade deployment is supported for SnapMirror volume replication only. It is not supported for SnapMirror qtree replication.



How SnapMirror works

The SnapMirror feature uses the information in the /etc/snapmirror.conf file and the information you enter via the `snapmirror.access` option or the /etc/snapmirror.allow file to establish a relationship between a specified *source* volume or qtree that you want to back up, and the *destination* volume or qtree where the backup is kept.

The SnapMirror feature does the following:

1. Creates a snapshot of the data on the source volume or qtree
2. Copies it to the destination, a read-only volume or qtree
3. Updates the destination to reflect incremental changes on the source, on the schedule you specify

The result of this process is an online, read-only volume or qtree that contains the same data as the source at the time of the most recent update.

What to use SnapMirror for

You might want to copy or use the data stored on a SnapMirror destination if you are in any of the situations described in "When to copy volumes" on page 246. The additional advantages of SnapMirror make it useful in other data retrieval situations, as described in the following table.

Situation	How to use SnapMirror
Disaster recovery: You want to provide immediate access to data after a disaster has made a qtree, volume, or filer unavailable.	You can make the destination writable so clients can use the same data that was on the source volume the last time data was copied.
Data restoration: You want to restore lost data on a qtree or volume source from its mirrored qtree or volume SnapMirror partner.	You can temporarily reverse the roles for the source and destination qtrees or volumes and copy the mirrored information back to its source.
Application testing: You want to use an application on a database, but you want to test it on a copy of the database in case the application damages the data.	You can make a copy of the database to be used in the application testing to ensure that the data on the source cannot be lost.

Situation	How to use SnapMirror
Load balancing: A large number of users need read-only access to a qtree or volume.	You can copy the data in a qtree or volume to multiple volumes or filers to distribute the load.
Off-loading tape backups: You need to reserve all processing and networking resources on a filer for serving NFS and CIFS requests.	After copying data on the source filer, you can back up the data in the destination to tape. This means that the source filer does not have to allocate resources for performing backups.
Remote access to data: Users who need read access to a volume are distributed over a large geographical area.	You can copy the source volume to other filers that are geographically closer to the users. Users accessing a local filer can read the data using less resource time than if they connected to a distant filer.

Differences between the vol copy command and SnapMirror

The `vol copy` command and the SnapMirror feature carry out similar functions. They both enable the administrator to copy snapshots of data from a source filer volume to a destination filer volume. However, they differ in several important ways.

- The vol copy feature does not require an additional license.
- The SnapMirror feature supports automated and scheduled updates of snapshot data mirrored between the source and destination filer volumes.
- The SnapMirror feature supports incremental snapshot updates between source and destination filer volumes.
- The SnapMirror feature supports qtree-level replication between the source and destination systems.

Differences between SnapMirror volume replication and qtree replication

You can configure SnapMirror replication for either whole volumes on a filer or individual qtrees on a volume.

SnapMirror volume replication: SnapMirror volume replication has the following characteristics:

- SnapMirror volume replication copies a snapshot of a source volume and all its qtrees to a destination.

- A destination volume set up for SnapMirror volume replication must first be set to restricted, read-only status.

SnapMirror qtree replication: SnapMirror qtree replication has the following characteristics:

- SnapMirror qtree replication copies only the contents of an individual qtree to a destination.
- If you need to mirror only the data stored on an individual qtree, then SnapMirror replication of that individual qtree uses less disk space on the filer.
- A destination qtree is read-only, but the volume on which it is located must be online and writable.
- SnapMirror qtree replication can be set up for a maximum of 255 qtrees on any one volume.

Maximum number of simultaneous replications on a filer

The maximum number of simultaneous SnapMirror replications that each filer model can support is shown in the following table.

Model	Number of Simultaneous Transfers Allowed
F85	4
F87	4
F720	4
F740	4
F760	4
F810	8
F820	8
F840	16
F880	16
F940	16
F960	16
R100	64

How transfers beyond the limit are handled: If more than the allowed number of SnapMirror volume or qtree replications are scheduled to run at the same time, each additional transfer will generate an error message stating that resource limits have been reached. Each transfer beyond the limit will re-attempt to run each following minute until it succeeds, SnapMirror is turned off, or the update is terminated.

SnapMirror commands and configuration files

About SnapMirror commands

This section lists the Data ONTAP commands and configuration files associated with the SnapMirror features that are described in this chapter.

Commands to set up SnapMirror

You can use the following commands and configuration files to set up SnapMirror replication.

Command or File	Function
`license add`	Enables the SnapMirror license on the source and destination filers. See "Entering license codes," on page 100.
`options snapmirror.access` or /etc/snapmirror allow file	Specifies the host names of filers that are allowed to copy data directly from the source filer. See "Specifying destination filers on the source filer," on page 89.
/etc/snapmirror.conf file	Specifies the destination volumes and qtrees to be replicated to and the schedule on which the destinations are updated. See "Defining source and destination through snapmirror.conf," on page 92.

`snapmirror on`
or
`options snapmirror.enable`
	Enables SnapMirror on the source and destination filers. See "Enabling SnapMirror," on page 100.

Command or File	Function

`vol create`
and
`vol restrict`
	In combination, create the restricted read-only volumes that are required as destinations for SnapMirror volume replications.
	See “Initializing a SnapMirror destination,” on page 102.

`snapmirror initialize`	Begins the initial complete (baseline) SnapMirror snapshot replication from a source volume or qtree to a destination volume or qtree. See “Initializing a SnapMirror destination,” on page 102 for more information.

SnapMirror management commands

You can use the following commands to perform SnapMirror management tasks:

Command or file	Function
`snapmirror update`	Performs a manual (unscheduled) update of the destination. See “Updating a destination manually,” on page 110.
`snapmirror status`	Displays the status of SnapMirror data transfers. See “Checking SnapMirror data transfer status,” on page 116.
/etc/log/snapmirror	Displays the SnapMirror data transfer history. See “Checking SnapMirror data transfer logs,” on page 125.
`snapmirror abort`	Stops a transfer that is in progress. See “Aborting a SnapMirror transfer,” on page 130.
/etc/snapmirror.conf	Enables you to specify or modify scheduled updates for one or more volumes or qtrees. See “Turning off or changing scheduled updates for volumes or qtrees,” on page 135.

SnapMirror shutdown commands

You can use the following commands to temporarily or permanently shut down SnapMirror processes on a filer, volume or qtree:

Command	Function

`snapmirror off`
or
`options snapmirror.enable off`
	Turns off SnapMirror functionality for a specified filer. See "Turning off SnapMirror updates for the filer," on page 137.

`snapmirror break`	Converts the destination to a writable volume or qtree, breaking the SnapMirror relationship between the source and destination. See "Converting a destination to a writable volume or qtree," on page 139.
`snapmirror release`	Releases for deletion SnapMirror snapshots on former source volumes or qtrees. See "Releasing partners from a SnapMirror relationship," on page 142.

SnapMirror advanced function commands

You can use the following commands to perform advanced functions with SnapMirror:

Command or File	Function
`snapmirror quiesce`	Stabilizes the contents of a destination just prior to a snapshot by allowing ongoing data transfers to finish and temporarily preventing new transfers from beginning. This action ensures a manual snapshot of a stable database. See "Stabilizing (quiescing) destinations before taking a snapshot," on page 132.
`snapmirror resume`	Resumes normal data transfer to a destination after it has been quiesced. See "Resuming transfers after quiescing a destination," on page 133.

Command or File	Function
`snapmirror resync`	Returns a former destination volume or qtree to the SnapMirror relationship after the `snapmirror break` command and resynchronizes its contents with the source without repeating the initial transfer. See “Resynchronizing SnapMirror,” on page 144.

/etc/snapmirror.conf file
`snapmirror initialize`
`snapmirror destinations.`
	These commands set up a cascading series of SnapMirror destinations. They make a uniform set of data available on a read-only basis to users from various locations throughout a network and they update that data uniformly at regular intervals. See “Copying from one destination to another in a series (cascading),” on page 151.

`snapmirror store`
and
`snapmirror use`
	These commands copy a volume to local tape and continue the backup on subsequent tapes if necessary. See “SnapMirroring a volume to local tape,” on page 156.

`snapmirror retrieve`
and
`snapmirror use`
	These commands initialize or restore a volume from local tape. See “Initializing a SnapMirror destination via local tape,” on page 160.

Considerations when planning and running SnapMirror

Prerequisites to running SnapMirror

The following prerequisites must be met before you can run SnapMirror:

- You must purchase and enable the SnapMirror license.

  If the SnapMirror source and destination are on different filers, you must purchase a license and enter the SnapMirror license code on each filer. For information on enabling the SnapMirror license, see "Entering license codes," on page 100.
- For SnapMirror volume replication, you must create a restricted volume to be used as the destination volume.

  SnapMirror does not automatically create a volume. See the information in the *System Administrator's Guide* in the section "Data Organization Using Volumes and Qtrees" for information about how to create volumes.
- For SnapMirror volume replication, the destination volume must run under a version of Data ONTAP equal to or later than that of the SnapMirror source volume. If you upgrade your filers to a newer version of Data ONTAP, upgrade the filers of SnapMirror destination volumes before you upgrade the filers of SnapMirror source volumes.
- For SnapMirror qtree replication, you must not create a qtree to be used as a destination qtree; the `snapmirror initialize` command creates the destination qtree automatically.
- For SnapMirror qtree replication, the destination qtree must run under Data ONTAP version 6.2 or later.
- The name and IP address of the source filer must be in the /etc/hosts file of the destination filer or must be resolvable by way of DNS or `yp`.

Restrictions

When planning SnapMirror configuration, consider the following restrictions:

- The source volume must be online.

  See the *System Administrator's Guide* for information about how to put a volume online.
- For SnapMirror volume replication, the capacity of the destination volume must be greater than or equal to the capacity of the source volume. See "Verifying the size of each volume" on page 250 for information about using the `vol status -b` command to check the capacity of the source volume and destination volume. See the *System Administrator's Guide* for

information about how to add disks to a volume. See “Initializing a SnapMirror destination,” on page 102 for information about disk sizes.

- To support SnapMirror qtree replication, the destination volume must contain 5% more free space than the source qtree consumes.
- The SnapMirror destination volume cannot be the root volume of a filer. The SnapMirror source volume, however, can be the root volume.
- A destination qtree can be on the root volume, but the /etc qtree cannot be a destination qtree.
- Destination qtrees names cannot
  - Consist of “*” or “-” or “/etc”
  - Contain the character “.” or the character combination “->”
  - Be longer than 64 characters
  - Be specified as “/vol/*volname*/” (with no qtree name)
  - Be specified as “*volname*/*qtree_name* (with no /vol/)”
- There must be a functional network to transfer data between two different filers.
- Each filer model supports a specified number of `snapmirror update` or `vol copy` operations at a time. (See “Maximum number of simultaneous replications on a filer” on page 77 for the number of copy operations each filer can support.)

Cautions

Be advised of the following cautions:

- Do not delete snapshots that SnapMirror creates in the source volume and then copies to the destination. The most recent SnapMirror snapshot is referred to as the newest common snapshot (NCS). Incremental changes to the destination depend on the newest common snapshot. If SnapMirror cannot find the required snapshot on the source, it cannot perform incremental changes to the destination.
- Do not use the `snapmirror release` or `snapmirror break` command on the destination volume or qtree unless you no longer need to copy incremental changes from the source. The destination must be actively functioning as a destination to receive incremental updates.
- Do not restrict or take the destination volume offline while SnapMirror is configured to transfer. Taking the destination offline prevents SnapMirror from performing updates to the destination.

Recommendations

Follow these recommendations to copy data efficiently:

- For smoothest performance, schedule SnapMirror snapshot and updates times (in the etc/snapmirror.conf file) to be different from the time of any regular system snapshots (set through the `snap_sched` command) on the source filer.
- For optimum SnapMirror volume replication performance, make sure that the SnapMirror source volume and destination volume contain disks of the same size, organized in the same RAID configuration.

  If the SnapMirror source and destination are qtrees, volume size and configuration do not make any difference.
- If you plan to do SnapMirror transfers through a firewall, you might need to know the port number used by SnapMirror. SnapMirror listens for connections on port 10566.

Ensuring accessible destinations via CIFS

Directories on all SnapMirror source volumes that support CIFS clients must be in Unicode format before being replicated to a destination; otherwise, the directories in the read-only destination volume will not be in Unicode format and attempts through CIFS to access directories and open files on the destination volume can receive “access denied” errors.

You can ensure that both source volume and destination volume directories are in Unicode format using the following methods:

- On the filer console for the source volume, enter the following two commands:
  - `vol options` *volname* `convert_ucode on` to convert any existing directories on the source volume to Unicode format.
  - `vol options` *volname* `create_ucode on` to automatically create any new directories on the source volume to Unicode format.
- Another option is to make sure that all directories on the source volume that will be accessed by CIFS clients are accessed by a CIFS client before initial replication to a destination. Such access on a writable source volume automatically converts that directory to Unicode format.

Setting up a basic SnapMirror operation

About a basic SnapMirror setup

This section describes the use of the SnapMirror feature for basic data replication between the SnapMirror source and destination, and basic recovery operations.

Note

If your source volumes contain directories that are accessed by CIFS clients, ensure that those directories are in Unicode format before carrying out the SnapMirror replication of that volume. See "Ensuring accessible destinations via CIFS," on page 85.

Step	Action
1	Make sure you have purchased a SnapMirror license for both the source and destination filers.
2	In both the source filer and the destination filer consoles, use the `license add` command to enable the SnapMirror license on the source and destination filers. **`license add` *`snapmirror_license_code`*** For more information see "Entering license codes," on page 100.
3	On the source filer console, use the `options snapmirror.access` command to specify the host names of filers that are allowed to copy data directly from the source filer. For example: **`options snapmirror.access host=d_filerA`** For more information see "How to specify destination filers on the source filer," on page 89.

Step	Action
4	Through the Data ONTAP AdminHost, create or edit the /etc/snapmirror.conf file on the destination filer to specify the volumes and qtrees to be copied and the schedule (*minute hour day_of_month day_of_week*) on which the destination is updated. For example, the following entry specifies snapshot mirroring from volume 0 of s_filerA to volume 1 of d_filerA at a maximum of 2000 kilobits per second every 15 minutes past every hour, Monday through Friday. **`s_filerA:vol0 d_filerA:vol1 kbs=2000,restart=always 15 * * 1,2,3,4,5`** For more information on schedule entries in the destination filer's /etc/snapmirror.conf file, see "Defining source and destination through snapmirror.conf," on page 92.
5	In both the source filer and destination filer console, use the `snapmirror on` command to enable SnapMirror on the source and destination filers. For more information, see "Enabling SnapMirror," on page 100.
6	Prepare the destination filer appropriately, depending on whether you are setting up SnapMirror volume or qtree replication. ◆ If you are setting up SnapMirror volume replication, in the destination filer console, use the `vol create` command to create a volume to be the destination for the snapmirrored volume on the source, then use the `vol restrict` command to mark the volume as restricted. ◆ If you are setting up SnapMirror qtree replication, make sure that the volume on the destination filer where you want to replicate a qtree with SnapMirror is online and not restricted. Do not manually create a destination qtree. For more information, see the information about creating volumes in the "Data Organization Using Volumes and Qtrees" section of the *System Administrator's Guide*; and see the information on restricting volumes in "How to initialize a SnapMirror destination," on page 102.

Step	Action
7	In the destination filer console, use the `snapmirror initialize` command to create an initial complete (baseline) copy of the source on the destination and start the mirroring process. **SnapMirror volume replication example:** Invoking the following command line transfers a complete copy of the source volume (vol0 on filerA) to the destination volume (vol2 on filerB). The destination volume must be configured as restricted and read-only. **`snapmirror initialize -S filerA:vol0 filerB:vol2`** **SnapMirror qtree replication example:** Invoking the following command line creates a destination qtree (qtree4 on vol1on filerB) and transfers a complete copy of the qtree source (qtree4 on vol1 on filerA) to that destination qtree. The volume in which the destination qtree is created must be online and writable. **`snapmirror initialize -S filerA:/vol/vol1/qtree4 filerB:/vol/vol1/qtree4`** After you invoke the `snapmirror initialize` command, the scheduled snapshot mirroring that you specified in Step 4 will automatically update the destination volume or qtree at the specified times. See "Initializing a SnapMirror destination," on page 102 for more information.
8	In the event that the SnapMirror source volume or qtree is disabled, you can use the `snapmirror break` command to make the destination volume or qtree writable and able to provide continuous file and data service to the clients who are no longer able to access the disabled source filer. For more information, see "Converting a destination to a writable volume or qtree," on page 139.

Specifying destination filers on the source filer

How to specify destination filers on the source filer

There are two methods of specifying destination filers on the source filer:

- You can specify the destination filers that are allowed access to the source filer by using the `snapmirror.access` option. This option specifies which SnapMirror destination filers can initiate transfers, and which network interfaces they can use.This is the preferred method for controlling SnapMirror access on a SnapMirror source filer.
- You can generate a snapmirror.allow file in the /etc/ directory on the source filer. The /etc/snapmirror.allow file specifies the host names of filers that are allowed to copy data directly from the source filer.

  If the `snapmirror.access` option is set to `legacy` (the default setting), the snapmirror.allow file defines the access permissions.

Specifying destinations using the snapmirror.access option

To specify the SnapMirror destinations allowed access to the SnapMirror source using the `snapmirror.access` option, complete the following step.

Step	Action
1	On the source filer, enter the following command: **`options snapmirror.access`** ***`access_specification`*** The syntax for specifying which filers are allowed access to the server is the same for SNMP, Telnet, and rsh, and is described in the na_protocolaccess(8) man page. For more information about the `options` command, see the na_options(1) man page. This option persists across reboots.

Example: If you want SnapMirror to copy data locally on filerA and to another filer named filerB, enter the following at the prompt on filerA:

```
filerA> options snapmirror.access host=filerA,filerB
```

Specifying destinations using the snapmirror.allow file

To specify the SnapMirror destinations allowed access to the SnapMirror source using the /etc/snapmirror.allow file, complete the following steps.

Step	Action
1	If the snapmirror.allow file does not already exist, use a text editor to create a file called snapmirror.allow in the /etc directory on the root volume of the source filer.
2	Add the name of each filer to which you are replicating data, on its own line, to the /etc/snapmirror.allow file. You do not have to add the name of the local filer.
3	Save edits to the file.

Example: If you want SnapMirror to copy data locally on filerA and to other filers named filerB and filerC, create a /etc/snapmirror.allow file on filerA with the following lines added:

```
filerB
filerC
```

Entries in the /etc/snapmirror.allow file are case-sensitive. You can use the `hostname` command on the destination filer to find the correct format for each entry in the /etc/snapmirror.allow file.

Resolving host names to their IP addresses

By default, the SnapMirror feature checks host names in the /etc/snapmirror.allow file against the host name sent from the destination filer. Alternatively, you can set SnapMirror to resolve the host names in the /etc/snapmirror.allow file to their IP addresses and compare them with the IP address of the destination filer.

The `snapmirror.checkip.enable` option controls how the host names are checked. When the option is off, which is the default, the entries in the /etc/snapmirror.allow file must match the host name of the destination filer reported by the `hostname` command. When the option is on, the source filer resolves the names in the snapmirror.allow file to IP addresses and then checks for a match with the IP address of the requesting destination filer. In this mode, literal IP addresses (for example, 123.45.67.89) and fully qualified names (for example, filerA.acme.com) can be valid entries in the /etc/snapmirror.allow file.

The /etc/snapmirror.allow file entry must map to the IP address of the originating network interface on the destination filer. For example, if the request comes from the IP address of a Gigabit Ethernet interface e10 named filerA-e10, then the /etc/snapmirror.allow file must contain “filerA-e10” or “filerA-e10.acme.com” so the name resolves to the correct IP address.

A local SnapMirror relationship, between two volumes on the same filer, does not require an entry in the /etc/snapmirror.allow file.

To configure SnapMirror to resolve host names to their IP addresses, complete the following step.

Step	Action
1	On the source filer, enter the following command: `options snapmirror.checkip.enable on`

Defining source and destination through snapmirror.conf

What the snapmirror.conf file does

The /etc/snapmirror.conf file defines the relationship between the source and the destination, the schedule used by the destination to copy data, and the arguments that control SnapMirror when copying data. This file resides on the destination filer.

How to distribute an snapmirror.conf file

You can create a single /etc/snapmirror.conf file for your site and copy it to all the filers that use SnapMirror. The /etc/snapmirror.conf file can contain entries pertaining to other filers. For example, the /etc/snapmirror.conf file on filerB can contain an entry for copying a volume from filerC to filerD. When filerB reads the /etc/snapmirror.conf file, it ignores the entries for other filers. Each time the file is read, a warning message is displayed for each line that is ignored.

When changes to the snapmirror.conf file take effect

If SnapMirror is enabled, changes to the /etc/snapmirror.conf file take effect within two minutes. If SnapMirror is not enabled, changes to the /etc/snapmirror.conf file take effect immediately after you enter the `snapmirror on` command to enable SnapMirror.

Creating and editing the snapmirror.conf file

To create or edit the /etc/snapmirror.conf file, complete the following steps.

Step	Action
1	If it does not already exist, use a text editor to create a file called snapmirror.conf in the /etc directory on the root volume of the destination filer.
2	If you want to add comments to the /etc/snapmirror.conf file, precede the comment with a pound sign (#). **Example:** `# Replicating from filerA`

Step	Action
3	For each destination volume or qtree that is to be located on this filer, type an entry specifying the source, destination, characteristics and schedule of the data transfer on one line using the following syntax: ***source_filer:{source_volume | /vol/volume_name/qtree_name}*** ***dest_filer:{dest_volume | /vol/volume_name/qtree_name}*** ***arguments schedule*** For a full description of the snapmirror.conf file entry syntax see "snapmirror.conf file entry syntax," on page 94. **Note** When you use a qtree as a source, you do not create another qtree to serve as the destination. SnapMirror automatically creates one for you, using the name you specify. However, you must specify the name of the qtree destination in the volume where you want it to be placed, either in the /etc/snapmirror.conf file or in the `snapmirror initialize` command. **Note** Do not specify more than 254 destination qtrees for any one volume in your /etc/snapmirror.conf file entries. If you want to combine all non-qtree data in a volume into a qtree to use as a SnapMirror source, use this syntax: ***source_filer:/vol/source_volume/-*** ***dest_filer:/vol/dest_volume/qtree_name*** The dash (-) character indicates all non-qtree data in the specified volume. **Note** The data in /vol/*source_volume*/– qtree can only be a SnapMirror source, never a destination. That is, after you create the /- qtree, you can copy data from it to another qtree you name, but you cannot copy data to it from another qtree.
4	Save edits to the file.

snapmirror.conf file entry syntax

The syntax for entries in the snapmirror.conf file is as follows:

```
src_filer:/vol/src_vol/[src_qtree]
dest_filer:/vol/dest_vol[/dest_qtree] [kbs=kbs],
[restart={never|always}] [sched]
```

The parameters in the entries are:

parameter	description
src_filer	The name of the filer from which you are copying data. SnapMirror uses the /etc/hosts file or the database used by DNS and NIS for name resolution. When SnapMirror searches for the source filer name, it should find the IP address for the source filer on the network over which you want the transfer to occur. **Example:** Assume you created a private network connecting a source filer and a destination filer and you named the interface on the source filer filerA-e0. The interface name filerA-e0 is what you enter in the *source_filer* field.
src_vol /*src_qtree*	The name of the volume or qtree that you are copying. Use the volume name alone for volumes. Use the full path name for qtrees. **Example:** If the name of the volume is vol1, enter vol1 in the *src_vol*/*src_qtree* field. If the name of the qtree is qtree3, and it is contained in volume vol3, enter the full path, /vol/vol3/qtree3, in the *src_vol*/*src_qtree* field.
dest_filer	The host name of the filer to which the data is copied. The name you use must be the exact host name of the destination filer. The *dest_filer* field is case-sensitive. You can use the `hostname` command on the destination filer to determine what you enter for this field. **Example:** If the name of the destination filer is filerA, enter filerA in the *dest_filer* field.

parameter	description
dest_vol [*/dest_qtree*]	The name of the destination volume or qtree to which you are copying data. Use the volume name alone for volumes. Use the full path name for qtrees. **Example:** If the name of the volume is vol1, enter vol1 in the *dest_vol*[*/dest_qtree*] field. If the name of the qtree is qtree4, and it is in vol2, enter the full path, /vol/vol2/qtree4, in the *dest_vol*[*/dest_qtree*] field.
kbs=*kbs*	Maximum transfer speed, in kilobytes per second, that Data ONTAP can use to transfer data The kbs and restart arguments are expressed as a comma-separated list of *name*=*value* pairs, with no spaces, for example: `kbs=2000,restart=always`
restart= {never | always | default}	Restart mode that SnapMirror uses to continue an incremental transfer from a checkpoint if it is interrupted. There are three options: ◆ Never—Transfers are always restarted from the beginning of a transfer and never from where they were before an interruption. This mode is useful if you must have the latest data on the destination. ◆ Always—Transfers are always restarted if possible from where they were before an interruption. This mode is useful for copying large volumes. ◆ Default—Transfers are restarted if they do not conflict with a scheduled transfer. This is the recommended option. SnapMirror always restarts from where the transfer was before an interruption. However, it does not restart if the restart occurs after a scheduled transfer, because the scheduled transfer gives more up-to-date data than the restarted transfer. The `kbs` and `restart` arguments are expressed as a comma-separated list of *name*=*value* pairs, with no spaces, for example: `kbs=2000,restart=always`

parameter	description

schedule	Determines the schedule the destination uses for updating data. The schedule is mandatory.
	
	The schedule consists of four space-separated fields in order:
	
	minute hour dayofmonth dayofweek
	
	minute can be a value from 0 to 59.
	
	hour can be a value from 0 (midnight) to 23 (11 p.m.).
	
	dayofmonth can be a value from 1 to 31.
	
	dayofweek can be a value from 0 (Sunday) to 6 (Saturday).
	◆ Multiple values, separated by commas, can be entered for any field.
	◆ All possible values for a field can be applied with an asterisk (*). If you specify an asterisk in each field of the schedule, SnapMirror updates the destination every minute.
	◆ A single dash (-) in any field means “never” and prevents this schedule entry from executing. (This is useful if you want the server to appear in the /etc/snapmirror.conf file so that `snapmirror update` can find it, but you do not want the SnapMirror scheduler to run automatically.)
	◆ A range of values for any field can be indicated with a low value and a high value separated by a dash. For example, you can indicate that you want an update every hour from 8:00 a.m. to 5:00 p.m by entering this value in the hour field:
	`8-17`
	◆ A range of values followed by a slash and a number indicates the frequency of the update. For example, you can indicate that you want an update every five minutes by entering this value in the minutes field:
	`0-59/5`

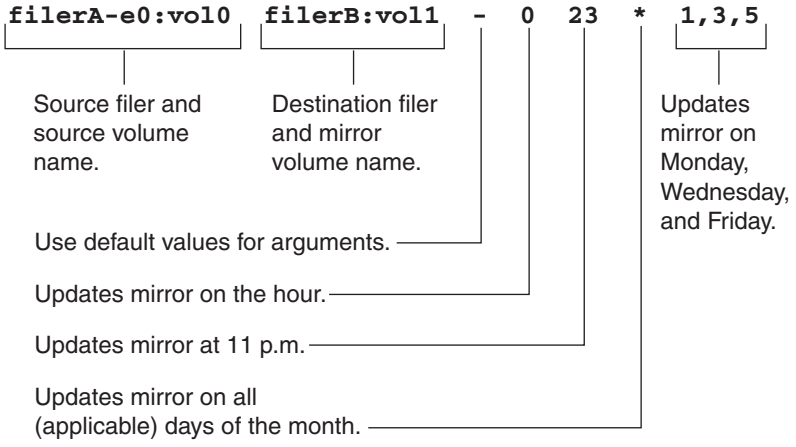
Example of snapmirror.conf schedule entries

Suppose you create a private network between filerA and filerB. In the /etc/hosts file on filerA, you give the host name for the interface as filerA-e0, and you ensure that filerA-e0 is also in the /etc/hosts file on filerB, the destination filer. You want to copy vol0 of filerA to vol1 of filerB over the private network every Monday, Wednesday, and Friday at 11 p.m. You also want to use the default for the arguments field.

To copy the data over the private network every Monday, Wednesday, and Friday at 11 p.m., you would enter the following in the /etc/snapmirror.conf file:

```
filerA-e0:vol0 filerB:vol1 - 0 23 * 1,3,5
```

The following figure illustrates what the entry in each field in the example means.



Additional snapmirror.conf examples

Example setting maximum update speed: The following line in an /etc/snapmirror.conf file sets the speed to 2000 kilobytes per second.

```
filerA:vol0 filerA:vol1 kbs=2000 15 * * 1,2,3,4,5
```

Note

The specified transfer speed might not be achievable because transfer speed is limited by factors such as network bandwidth.

Example specifying always restart: The following line in an /etc/snapmirror.conf file sets the restart value to `always`.

```
filerA:vol0 filerA:vol1 kbs=2000,restart=always 15 * * 1,2,3,4,5
```

Example specifying default values for maximum speed and restart:

If you set the value of only one argument (`kbs` or `restart`), the other will use the default value. If you want to use the default argument for both values, enter a dash (-).

The following line in an /etc/snapmirror.conf file sets both arguments to the default value.

```
filerA:vol0 filerA:vol1 - 15 * * 1,2,3,4,5
```

Example specifying 15-minute interval updates during specific hours: If you want to schedule an update every afternoon at 1:00, 1:15, 1:30, 1:45, 5:00, 5:15, 5:30, 5:45, 7:00, 7:15, 7:30, and 7:45, you enter the following in the schedule field:

```
filerA:vol0 filerA:vol1 0,15,30,45 13,17,19 * *
```

Note

An update is started when the current time matches a value in all four fields. Be careful that a value in the day of the month field does not exclude a value in the day of the week field. For example, the following schedule would update the destination every afternoon at 1:00, 1:30, 3:00, 3:30, 5:00 and 5:30 but only when the first day of the month falls on a Monday.

```
filerA:vol0 filerA:vol1 0,30 13,15,17 1 1
```

Note

The schedule represents the goal of the SnapMirror feature. Factors that may prevent SnapMirror from updating every minute include resource limitations or network stability. If an update is in progress when another is scheduled to occur, SnapMirror will start another transfer as soon as the first is complete. However, if three updates pass while the current transfer is in progress, SnapMirror does only one more update; it does not go back and run updates that have been made obsolete by those scheduled later.

What restarts and retries are

In SnapMirror, a *retry* is an automatic attempt to start the transfer process after an interruption, whether or not any data was successfully transferred. A *restart* is the resumption of a previous transfer process from a restart checkpoint.

If a transfer fails, because of network failure, for example, SnapMirror automatically *retries* the transfer. SnapMirror makes a restart checkpoint every five minutes during a transfer. If a restart checkpoint exists and conditions are right, SnapMirror *restarts* the previous transfer where it left off. If not, SnapMirror creates a new snapshot and starts a new transfer.

After a reboot, SnapMirror does not automatically retry a transfer that was interrupted, but the next scheduled or manual transfer restarts it at the restart checkpoint, if the checkpoint is still valid.

A restart can happen if all of the following conditions are present:

- A restart checkpoint exists.
- All snapshots being transferred still exist.
- The value for restart mode in the /etc/snapmirror.conf file is set to *always* or is not set, and the next scheduled update has not arrived.

An initial transfer can be restarted but will not be retried automatically. To restart an initial transfer, you need to re-enter the `snapmirror initialize` command. Scheduled incremental updates automatically retry the transfer. A manual update issued from the command line is not retried automatically.

Enabling SnapMirror

SnapMirror enabling procedures

Before any SnapMirror replication process can begin, the SnapMirror feature must be licensed and enabled on the filer.

Entering license codes

Enter the SnapMirror license code on the filer that is to be the source and on the filer that is to be the destination.

To enter the SnapMirror license code, complete the following step.

Step	Action
1	On the server, enter the following command: **`license add xxxxxxx`** *xxxxxxx* is the license code you purchased. This setting persists across reboots.

Turning SnapMirror on

To turn SnapMirror on, complete the following steps.

Step	Action
1	Enter the following command on both the source filer and destination filer to enable SnapMirror: **`options snapmirror.enable on`** Alternatively, you can still use the older command instead to turn SnapMirror on: **`snapmirror on`** This setting persists across reboots.

Step	Action	
2	**If you are using the snapmirror.access option to specify allowed destinations...**	**If you are using the /etc/snapmirror.conf file to specify allowed destinations...**
	On the source, enter the following command as a single line: **`options snapmirror.access host=[dest_filer1,dest_filer2...]`** The default value for the `snapmirror.access` option is `legacy`, which lets the /etc/snapmirror.allow file define the access permissions. This option persists across reboots.	Add the name of each destination filer, on its own line, to the /etc/snapmirror.allow file.

Initializing a SnapMirror destination

When to run a complete SnapMirror transfer

You must use the `snapmirror initialize` command to perform a complete (baseline) transfer of information whenever you start up a SnapMirror source-destination relationship for the first time. This process is known as initializing a destination.

How quotas apply to destination qtrees

Qtree quotas apply to qtrees in a SnapMirror relationship. If a destination qtree is limited to 100 GB, transfers from a source greater than 100 GB will fail until the source drops to less than 100 GB or the qtree quota is increased on the destination.

User quotas also apply to SnapMirror destination qtrees; however, a SnapMirror qtree update will not fail if user quotas are exceeded.

Initializing a SnapMirror destination volume from tape

You can also initialize a destination volume from tape using the `snapmirror store` command on the source volume and the `snapmirror retrieve` command on the destination volume. See "SnapMirroring a volume to local tape," on page 156 and "Initializing a SnapMirror destination via local tape," on page 160 for more information. The `snapmirror store` and `snapmirror retrieve` functions are valid only for volumes, not for qtrees in a SnapMirror relationship.

How to initialize a SnapMirror destination

To initialize a SnapMirror destination, complete the following steps.

Note

If your source volumes contain directories that are accessed by CIFS clients, ensure that those directories are in Unicode format before carrying out the initial SnapMirror replication of that volume. See "Ensuring accessible destinations via CIFS," on page 85.

Step	Action
1	If the destination is a volume, is online, and has not been initialized before, enter the following command from the destination filer: **`vol restrict dest_volume`** *dest_volume* is the destination volume. **Note** Do not use the `vol restrict` command on a qtree. If you are initializing a qtree, go to step 2.

Step	Action
2	From the destination filer, enter the following command: **`snapmirror initialize [`*`options`*`] [`*`dest_filer:`*`] {`*`dest_volume`*` | `*`qtree_path`*`}`** *options* can be one or more of the following: ◆ `-k` *n* sets the maximum transfer speed to *n* kilobytes per second. This option has the same effect as the `kbs` argument in the /etc/snapmirror.conf file. ◆ `-S` [*source_filer*:]{*source_volume* | *source_qtree_path*} specifies the source filer and volume or qtree to copy. *source_volume* is the volume you want to copy. The source specified must match an entry for *source_volume* in the /etc/snapmirror.conf file, if one exists. If an entry exists but does not match, the operation displays an error message and terminates. If there is no entry for the specified source, the command runs. *source_qtree_path* is the path to the qtree you want to copy. If the `-S` option is not set, the source must be specified in the /etc/snapmirror.conf file. If it is not specified, the operation displays an error message and terminates. **Note** The *source_qtree_path* can be a qtree in a SnapMirror destination volume. ◆ `-c` *snapshot_name* creates a snapshot (with the name *snapshot_name*) of a qtree on the destination after the next update (so that it does not compete with any ongoing updates). SnapMirror does not lock or delete this snapshot. *snapshot_name* cannot be minutely.x, hourly.x, nightly.x, or weekly.x, because these names are reserved for scheduled snapshots. This option is valid only for a qtree. ◆ `-s` *snapshot_name* specifies an existing source qtree snapshot to be transferred. This prevents the normal action of the source creating a snapshot to transfer. SnapMirror does not lock or delete this snapshot. This option is valid only for a qtree SnapMirror replication.

Step	Action
	dest_filer is the name of the destination filer. The destination can reside on the same filer as the source or on another filer. *dest_volume* or *qtree_path* specifies the destination volume or qtree. If it is associated with a local source specified in the /etc/snapmirror.conf file, SnapMirror uses that source If the destination volume or qtree specified is not in a scheduled relationship, then the `-S` option must be used to provide a source. The `snapmirror initialize` command creates the destination qtree, but you must specify the destination qtree name at the end of the path as though it already existed. If the destination qtree exists before the command runs, the command fails.

Example 1:

```
filerB> snapmirror initialize -S filerA:vol0 filerB:vol2
```

Results:

SnapMirror transfers a complete copy of the source volume (vol0 on filerA) to the destination volume (vol2 on filerB).

Example 2:

```
filerB> snapmirror initialize -S filerA:/vol/vol1/qtree4
filerB:/vol/vol1/qtree4bak
```

Results:

SnapMirror transfers a complete copy of the qtree source (qtree4 on vol1 on filerA) to the destination qtree (qtree4bak on vol1on filerB).

Initializing a destination for non-qtree data

Non-qtree data is any data on a filer that is not contained in its qtrees. Non-qtree data can include:

- Configuration and logging directories on the filer (for example, /etc or /logs) that are not normally visible to filer clients
- Directories and files on a filer volume that has no qtree configured

To use SnapMirror to replicate non-qtree data from a source to a destination, complete the following step.

Step	Action
1	From the destination filer, enter the following command: **snapmirror initialize -S *source_filer*:/vol/*source_volume*/- *dest_filer*:/vol/*dest_volume*/*qtree_name*** The dash (-) character indicates all non-qtree data in the specified volume. The `snapmirror initialize` command transfers the non-qtree data in the volume you specify, but you must specify the destination qtree name at the end of the path as though it already existed. If the destination qtree exists before the command runs, the command fails.

Example:

```
filerB> snapmirror initialize -S filerA:/vol/vol3/-
filerB:/vol/vol4/non_qtree_data_in_vol3
```

Results:

SnapMirror transfers to the destination qtree (non_qtree_data_in_vol3 on vol4 on filerB) a complete copy of all the data in filerA, volume 3, that is not a part of a qtree.

After the transfer, the non-qtree data can only be a SnapMirror source, never a destination. Although you can copy data from the non-qtree data to another qtree, you cannot copy data to the non-qtree data from another qtree.

If you run the `snapmirror quiesce` or the `snapmirror break` command on the destination volume (/vol/vol4/non_qtree_data_in_vol3), you can resynchronize the destination volume to the source volume, as in this example:

```
filerB> snapmirror resync -S /vol/vol3/-
/vol/vol4/non_qtree_data_in_vol3
```

You cannot resynchronize the two qtrees in the opposite direction.

To summarize, you cannot make /vol/volname/- a SnapMirror destination. This non-qtree data can only be a SnapMirror source.

How snapmirror initialize copies volumes

When the `snapmirror initalize` command copies a volume, it creates a snapshot of all the data on the source and transfers it to the destination. The destination is a volume you have already created and marked restricted. After SnapMirror finishes transferring the data, it brings the destination online, but in a read-only state. This version of the destination is the baseline for the first incremental update.

Note

While the initial data transfer is taking place, the destination is marked "invalid" in the output of a `vol status` command. The volume becomes valid and goes online once the initial transfer is complete.

How snapmirror initialize copies qtrees

To use SnapMirror to copy a qtree, you do not create a destination qtree because the `snapmirror initialize` command creates it. The volume where you want the destination qtree to be must be online. After the destination qtree is initialized, it is no longer writable; however, the rest of the volume where that qtree resides is still writable.

The destination snapshot created by qtree initialization is marked "busy" in the output of the `snap list` command until the next transfer is complete.

What happens after SnapMirror makes the initial copy to the destination

After you initialize a SnapMirror volume replication, the files and snapshots in the source volume are available on the destination.

After you initialize a SnapMirror qtree replication, the files on the source qtree are available on its destination qtree.

You can export the destination for NFS mounting or add a share corresponding to the destination for CIFS sharing.

How to check on the initialization of a volume

To check that a destination volume has been initialized, you can use the `snapmirror status` command. For more information, see "Checking SnapMirror data transfer status," on page 116. If you specify no options or arguments, the `snapmirror status` command shows the status of the volumes in the filer, as shown in the example below. You also can use the `vol status` or the `qtree` command to check whether the volume or qtree is a SnapMirror destination.

```
filerA> snapmirror status
Snapmirror is on.
```

```
Source          Destination     State          Lag       Status
filerA:vol0     filerA:vol0bak  Snapmirrored   00:56:58  Idle
filerA:vol1     filerB:vol6     Source         23:69:26  Transferring
(126 MB done)
```

How to check on the initialization of a qtree

To check that a destination qtree has been created and initialized, you can use the `qtree` command. If you specify no options or arguments, the `qtree` command shows the status of all the qtrees in the filer, as in the example below.

```
filerA> qtree

Volume          Tree    Style Oplocks    Status
--------        -----   ----- -------    ------
vol0                    unix  enabled    normal
qtree24                 unix  enabled    normal
filerB_vol0             unix  disabled   normal
filerB_vol0     qt1     mixed enabled    snapmirrored
filerB_vol0     qt2     unix  disabled   normal
filerB_vol0     qt3     ntfs  enabled    snapmirrored
```

Note

When the `qtree` command is run from a MultiStore™ vfiler™ unit, any qtree that is either a SnapMirror destination or part of a volume that is a SnapMirror destination has read_only in the Status column of the output.

How snapmirror initialize matches source and destination volume size

When you use the `snapmirror initialize` command to initialize a volume replication, SnapMirror sets the `vol options fs_size_fixed` option to `on`. This option forces the file system on the destination volume to remain the same size as the file system on the source volume.

What you can do if initialization fails or is interrupted

When an initialization fails because of a reboot or other interruption, you can restart it by re-entering the `snapmirror initialize` command if all the following conditions are present:

- The output of the `snapmirror status` command shows that the process has a restart checkpoint.
- The snapshots being transferred still exist.
- The disk geometry has not changed.

- The value for restart mode in the /etc/snapmirror.conf file is set to *always* or is set to the default and the next scheduled update has not begun.

SnapMirror does not automatically retry to initialize a destination.

Updating a destination manually

Why to run a manual incremental SnapMirror update

Ordinarily, SnapMirror updates the destination automatically according to the update schedule you specify in the /etc/snapmirror.conf file. However, you can also initiate updates manually with the `snapmirror update` command. You might need to run an unscheduled update to prevent data loss resulting from a scheduled or threatened power outage or from a destination volume being taken offline for maintenance, repair, upgrade, or data migration. You can also include the `snapmirror update` command in an external script if you want to drive updates using that script.

How to perform a manual SnapMirror update

To perform an unscheduled SnapMirror incremental update, independent of the schedule in the /etc/snapmirror.conf file, complete the following step.

Step	Action
1	From the destination filer, enter the following command: **`snapmirror update [`*`options`*`] [`*`dest_filer`*`:] {`*`dest_volume`*` | /vol/`*`dest_volume/qtree_path`*`}`** *options* can be one or more of the following: ◆ `-k` *n* sets the maximum transfer speed to *n* kilobytes per second. This option has the same effect as the `kbs` argument in the /etc/snapmirror.conf file. ◆ `-s` *snapshot_name* specifies an existing (qtree only) source snapshot to be transferred, rather than a snapshot taken by the source. SnapMirror does not lock or delete this snaphot. *snapshot_name* cannot be minutely.x, hourly.x, nightly.x, weekly.x, snapshot_for_backup.x or snapshot_for_volcopy.x. You must rename such snapshots on the source and then copy them.

Step	Action
	◆ `-c` *snapshot_name* creates a snapshot named *snapshot_name* of a qtree on the destination after the next update (so that it does not compete with any ongoing updates). SnapMirror does not lock or delete this snapshot. *snapshot_name* cannot be minutely.x, hourly.x, nightly.x, or weekly.x, because these names are reserved for scheduled snapshots. ◆ `-S` *[source_filer:]source_volume* | *qtree_path* specifies the source filer and volume for the update. *source_volume* is the volume you want to copy. The source specified by the `-S` option must match an entry for *source_volume* in the /etc/snapmirror.conf file. If one exists but does not match, the operation displays an error message and terminates. If there is no entry for the specified source volume, the command runs. If the `-S` option is not set, the source must be specified in the /etc/snapmirror.conf file. If it is not specified, the operation displays an error message and terminates. *dest_filer* specifies the name of the destination filer. *dest_volume* specifies the destination volume. If it is a scheduled destination of a local source volume as specified in the /etc/snapmirror.conf file, that source volume is assumed to be the source. If the destination volume specified is not in a scheduled relationship, then the `-S` option must be used to provide a source.

Example 1: `filerB> snapmirror update filerB:vol2`

Result: SnapMirror updates the destination (vol2 on filerB) from the source specified in the /etc/snapmirror.conf file.

Example 2: `filerB> snapmirror update -S filerA:/vol/vol1/users filerB:/vol/vol2/usersbak`

Result: SnapMirror updates the qtree destination on filerB:/vol/vol2/usersbak from the source qtree on filerA:/vol/vol1/users.

Creating extra backup snapshots for SnapMirror qtrees

To establish an extra backup snapshot on both sides of a SnapMirror qtree relationship, you can create a manual snapshot using the `snap create` command and then use the `-c` and `-s` options of the `snapmirror update` command together. These extra backup snapshots can serve as the newest common snapshot in case a base snapshot was accidentally deleted. You can also use them to resynchronize a SnapMirror qtree relationship to an earlier resynchronization point.

Example:

```
filerB> snap create vol2 my_snap

filerA> snapmirror update -S filerB:/vol/vol2/qtree1 -s my_snap -c
my_dest_snap vol/vol4/qtreeSafe
```

Result: You create a snapshot on filerB (the source) called my_snap. Then on filerA (the destination), you update the data in filerA:/vol/ vol4/qtreeSafe from the data in my_snap and store the updated data in the snapshot you create called my_dest_snap. SnapMirror does not automatically delete my_snap and my_dest_snap because they are user-originated snapshots.

What happens after SnapMirror makes incremental updates to the destination

Changes on the source are reflected on the destination after SnapMirror completes the transfer. Changes being copied are not visible during the transfer and are not visible if the transfer is interrupted. After the destination update is finished, you see the changes if you open the file.

SnapMirror automatically deletes old snapshots that are no longer necessary for updating data.

Listing SnapMirror snapshots

About SnapMirror listing

You can use the Data ONTAP `snap list` command to list all snapshots, including the SnapMirror-specific snapshots, that are stored on your filer.

To list all the snapshots on your system, complete the following step.

Step	Action
1	In the console of either your source or destination filer, enter the following command: **`snap list vol_name`** **Result:** A listing of all snapshots stored on your system appears. SnapMirror snapshots are distinguished from system snapshots by a more elaborate naming convention and the label “snapmirror” in parentheses.

Naming conventions for snapshots used by SnapMirror

When you run the `snap list` command, SnapMirror snapshots are distinguished from the regular filer system snapshots by their naming conventions.

For volume replication, SnapMirror creates a snapshot of the whole source volume that is copied to the destination volume. For qtree replication, SnapMirror creates snapshots of one or more source qtrees on the source volume that are copied to a qtree on the destination volume.

A SnapMirror volume snapshot name is in the following format:

dest_filer(sysid)_name.number

dest_filer is the host name of the destination filer.

sysid is the destination system ID number.

name is the name of the destination volume.

number is the number of successful transfers for the snapshot, starting at 1. Data ONTAP increments this number for each transfer.

A SnapMirror qtree snapshot name is in the following format:

dest_filer(sysid)_name-src|dst.number

dest_filer is the host name of the destination filer.

sysid is the destination system ID number.

name is the name of the destination volume or qtree path.

src|dst is the source or destination name.

number is an arbitrary start point number for the snapshot. Data ONTAP increments this number for each transfer.

In the output of the `snap list` command, SnapMirror snapshots are followed by the SnapMirror name in parentheses.

Volume example:

```
filerA(0016791363)_vol0.9 (snapmirror)
```

Qtree example:

```
filerA(0016789302)_vol1_qtree3-dst.15 (snapmirror)
```

Caution

Do not delete manually created snapshots marked “snapmirror” in the output of the `snap list` command.

Using snap list to show SnapMirror updates on the destination volume

The following example describes SnapMirror snapshots that are created on a source volume and copied to a destination volume. In this example, data is copied from vol1 of filerA (the source) to vol2 of filerB (the destination).

To create a baseline version of a destination volume, filerA creates a snapshot named filerB(0016782130)_vol2.1 on filerA. All snapshots in vol1 of filerA, including filerB(0016782130)_vol2.1, are transferred to vol2 of filerB. When replicating a qtree, SnapMirror transfers only the qtree’s data in the snapshot for the qtree.

If the administrator were to run the `snap list` command on the destination filerB after the filerB(0016782130)_vol2.1 snapshot was transferred from filerA to filerB, a listing similar to the following example would be generated.

```
filerB> snap list vol2
working.....

%/used    %/total  date          name
--------  --------  ------------  --------
0% ( 0%) 0% ( 0%)  Nov 17 10:50  filerB(0016782130)_vol2.1
(snapmirror)
```

```
1% ( 0%) 0% ( 0%)  Nov 17 10:00  hourly.0
1% ( 0%) 0% ( 0%)  Nov 17 00:00  nightly.0
1% ( 0%) 0% ( 0%)  Nov 15 16:00  hourly.1
1% ( 0%) 1% ( 0%)  Nov 15 15:00  hourly.2
2% ( 0%) 1% ( 0%)  Nov 15 14:00  hourly.3
2% ( 0%) 1% ( 0%)  Nov 15 13:00  hourly.4
2% ( 0%) 1% ( 0%)  Nov 15 12:00  hourly.5
```

When it is time to update the destination, another snapshot is created on filerA.

The `snap list` command on filerA would generate the following display after the filerB(0016782130)_vol2.2 snapshot was created on filerA:

```
filerA> snap list vol1
working....
%/used   %/total     date          name
-------- ----------  ------------  --------
0% ( 0%) 0% ( 0%)    Nov 17 10:52  filerB(0016782130)_vol2.2
(snapmirror)
0% ( 0%) 0% ( 0%)    Nov 17 10:51  filerB(0016782130)_vol2.1
(snapmirror)
1% ( 0%) 0% ( 0%)    Nov 17 10:00  hourly.0
1% ( 0%) 0% ( 0%)    Nov 17 00:00  nightly.0
1% ( 0%) 0% ( 0%)    Nov 15 16:00  hourly.1
1% ( 0%) 1% ( 0%)    Nov 15 15:00  hourly.2
```

After the filerB(0016782130)_vol2.2 snapshot is transferred from filerA to filerB, both snapshots exist on filerB. On filerA, however, filerB(0016782130)_vol2.1 is no longer needed and is deleted; only filerB(0016782130)_vol2.2 is retained to be used for the next transfer.

You can see a list of each SnapMirror snapshot on the server, and the qtrees it contains, and the client sources of those qtrees and their timestamps by using the `snap list -q` command.

You can use the `snap list -o` command to display the names, timestamps, and sources (if they are copies) of the qtrees in a specified volume or at a path name.

Checking SnapMirror data transfer status

Why to check data transfer status

You check data transfer status, using the `snapmirror status` command, to determine the status of all existing SnapMirror relationships on the filer.

What checking status shows you

If you check the status and you enabled SnapMirror, messages of the following form are displayed:

```
filerA> snapmirror status
Snapmirror is on.
Source                  Destination             State          Lag       Status
filerA:vol0             filerA:vol1             Snapmirrored   02:25:11  Transferring
(60 MB done)
filerB:/vol/vol1/qtree  filerB:/vol/vol3/qtree  Quiesced       00:01:15  Idle
```

You see a status report for any SnapMirror source that contains the base snapshot, and for any destination in a current SnapMirror relationship or listed in the /etc/snapmirror.conf file. Destinations that were broken through the `snapmirror break` command but still contain the base snapshot are listed.

If you check the status of data transfer and you did not enable SnapMirror, the following message is displayed:

```
filerA> snapmirror status
Snapmirror is off.
```

The status of SnapMirror relationships, if any, are still displayed, as shown in the example above.

Checking the status

To check the status of data copying and check on how current the information at a destination is, complete the following step.

Step	Action
1	Enter the following command: **`snapmirror status [options] [[filer:] [path] ...]`** *options* can be one of the following: ◆ `-l` displays the long format of the output, which contains more detailed information.See Example 2, below. ◆ `-q` displays which volumes or qtrees are quiesced or quiescing. For more information, see Example 3, below, and "Stabilizing (quiescing) destinations before taking a snapshot," on page 132. *filer* is the name of the source filer. *path* is the name of the source volume or the path to and name of the source qtree. **Result:** If no arguments or options are given, SnapMirror displays a message showing whether a transfer is in progress, how much of the data transfer has been completed, the state of the destination, and the amount of time since the last snapshot was created and transferred successfully, as shown in Example 1.

Examples

The following examples describe how the `snapmirror status` command displays the status of a data transfer. See the table "What the categories in the command output mean," on page 120 for more information.

Example 1: With no options, the information displayed by the `snapmirror status` command might look like this:

```
filerB> snapmirror status
Snapmirror is on.
Source                 Destination            State          Lag        Status
filerA:vol0            filerB:vol2            Broken-off     29:09:58   Idle
filerC:vol0            filerB:vol3            Snapmirrored   00:09:53   Idle with
restart checkpoint (23 MB done)
filerC:vol4            filerB:vol5            Snapmirrored   00:04:58   Transferring
(36 MB done)
filerA:/vol/vol1/qt5   filerB:/vol/vol4/qt5   Quiesced       00:05:12   Idle
filerC:/vol/vol2/qt1   filerB:/vol/vol1/qt2   Snapmirrored   00:02:33   Quiescing
```

Example 2: With the `-l` option, the configuration described in Example 1 would look like this.

```
filerB> snapmirror status -l

Snapmirror is on.
Source:                 filerA:vol0
Destination:            filerB:vol2
Status:                 Idle
Progress                -
State:                  Broken-off
Lag:                    29:09:58
Mirror Timestamp:       Sat Jul 15 00:50:02 GMT 2000
Base Snapshot           tpubs-f720(0016791363)_vol2.1249
Current Transfer Type:  -
Current Transfer Error: -
Contents:               -
Last Transfer Type:     Update
Last Transfer Size:     122000 KB
Last Transfer Duration: 00:01:02
Last Transfer From:     filerA:vol0

Source:                 filerC:vol0
Destination:            filerB:vol3
Status:                 Idle with restart checkpoint
Progress                23552 KB done
State:                  Snapmirrored
Lag:                    00:09:53
Mirror Timestamp:       Sun Jul 16 05:50:07 GMT 2000
Base Snapshot:          filer2(0016778780)_vol3.985
Current Transfer Type:  -
Current Transfer Error: Abort by user
Contents:               Replica
Last Transfer Type:     Update
Last Transfer Size:     432000 KB
Last Transfer Duration: 00:01:23
Last Transfer From:     filerC:vol0

Source:                 filerC:vol4
Destination:            filerB:vol5
Status:                 Transferring
Progress                36864 KB done
State:                  Snapmirrored
Lag:                    00:04:58
Mirror Timestamp:       Sun Jul 16 05:55:02 GMT 2000
Base Snapshot:          filerB(0016778780)_vol5.57843
```

```
Current Transfer Type:  Scheduled
Current Transfer Error: -
Contents:               Replica
Last Transfer Type:     Scheduled
Last Transfer Size:     345000 KB
Last Transfer Duration: 00:03:23
Last Transfer From:     filerB:vol4

Source:                 filerC:/vol/vol1/qt5
Destination:            filerB:/vol/vol4/qt5
Status:                 Idle
Progress                -
State:                  Quiesced
Lag:                    0:05:12
Mirror Timestamp:       Sun Jul 16 05:56:12 GMT 2000
Base Snapshot:          filerB(0016778780)_vol_vol4_qt5.54
Current Transfer Type:  -
Current Transfer Error: -
Contents:               Replica
Last Transfer Type:     Scheduled
Last Transfer Size:     45000 KB
Last Transfer Duration: 0:00:12
Last Transfer From:     filerC:/vol/vol1/qt5

Source:                 filerC:/vol/vol2/qt1
Destination:            filerB:/vol/vol4/qt2
Status:                 Quiescing
Progress                -
State:                  Snapmirrored
Lag:                    0:02:33
Mirror Timestamp:       Sun Jul 16 05:58:20 GMT 2000
Base Snapshot:          filerB(0016778780)_vol_vol4_qt2.122
Current Transfer Type:  -
Current Transfer Error: -
Contents:               Transitioning
Last Transfer Type:     Scheduled
Last Transfer Size:     80 KB
Last Transfer Duration: 0:00:08
Last Transfer From:     filerC:/vol/vol2/qt1
```

Example 3: With the `-q` option, the output might look like this.

```
filerC> snapmirror status -q
Snapmirror is on.
vol3 is quiesced
vol2 has quiesced/quiescing qtrees:
         /vol/vol2/qt1 is Quiescing
```

```
/vol/vol2/qt2 is Quiesced
```

What the categories in the command output mean

Information messages that the `snapmiror status` command can display are as follows.

snapmirror status Source entries:

Source entry	Description
filer:vol	The source filer and source volume
filer:qtree_path	The source filer and qtree path
-	Either the SnapMirror destination is an imported volume without an entry in the /etc/snapmirror.conf file, or a Data ONTAP upgrade is in progress.
filer:tape_device	The source tape device; transfer from this tape device is still in progress
base snapshot	The name of the base snapshot from which a completed transfer was made, if the source is a tape device and there is no entry in the /etc/snapmirror.conf file for the destination

snapmirror status Destination entries:

Destination entry	Description
filer:vol	The destination filer and volume
filer:qtree_path	The destination filer and qtree path
filer:tape_device	The destination tape device; transfer to this tape device is in progress
tape_destination	Displayed after a transfer to tape is finished. The `snapmirror destinations` command also displays this information. For more information, see "Listing SnapMirror destinations for a volume in a cascading series," on page 152.

State entry	Description
Uninitialized	The destination is listed in the /etc/snapmirror.conf file, but the volume or qtree has not been initialized or the destination is being initialized.
Snapmirrored	The volume or qtree is in a SnapMirror relationship.
Broken-off	The destination was in a SnapMirror relationship, but a `snapmirror break` command made the volume or qtree writable. This state is reported as long as the base snapshot is still present in the volume. If the snapshot is deleted, the state is listed as "uninitialized" if the destination is in the /etc/snapmirror.conf file or is no longer listed if it is not. A successful `snapmirror resync` command restores the snapmirrored status.
Quiesced	SnapMirror is in a consistent internal state and no SnapMirror activity is occurring. In this state, you can create snapshots with confidence that all destinations are consistent. The `snapmirror quiesce` command brings the destination into this state. The `snapmirror resume` command restarts all SnapMirror activities.
Unknown	The destination volume or the volume that contains the destination qtree is in an unknown state. It might be offline or restricted.
Source	When the `snapmirror status` command is run on the source filer and the destination is on another filer, the state of the destination is unknown, so the source status is reported.

Lag entry	Description
hh:mm:ss	Indicates the difference between the current time and the timestamp of the snapshot last successfully transferred to the destination.

Lag entry	Description
-	The destination is not initialized.

snapmirror status Status entries:

Status entry	Description
Idle	No data is being transferred.
Idle with restart checkpoint (*n X*B done)	No data is being transferred. The last transfer attempt was aborted, but the transfer saved a restart checkpoint and thus can be restarted at the next attempt. Transfer sizes are reported in KB up to 10,240, MB up to 10,240, GB up to 10,240, then TB.
Transferring	Transfer has been initiated but has not yet started, or is just finishing.
Transferring (*n X*B done)	Data transfer is in progress. Transfer sizes are reported in KB up to 10,240, then MB up to 10,240, then GB up to 10,240, then TB.
Pending	The destination was not updated because of a transfer failure; the transfer will be retried automatically.
Pending with restart checkpoint (*n X*B done)	The destination was not updated because of a transfer failure. The transfer will be retried automatically from the restart checkpoint. Transfer sizes are reported in KB up to 10,240, MB up to 10,240, GB up to 10,240, then TB.
Aborting	A transfer is being aborted and cleaned up.
Quiescing	The specified volume or qtree is waiting for all existing transfers to complete. The destination is being brought into a stable state.
Resyncing	The specified volume or qtree is being matched with data in the common snapshot.
Waiting	SnapMirror is waiting for a new tape to be put in the tape device.

snapmirror status additional entries:

Additional -l option entries	Description
Progress	Shows the amount of data (in KB) transferred by the current transfer. Shows the restart check point if the status is Idle or Pending.
Mirror Timestamp	The timestamp of the last snapshot successfully transferred from the source to the destination. **Note** A resynchronization may change the base snapshot to a snapshot with an older timestamp.
Base Snapshot	The name of the base snapshot for the destination. For volumes in a SnapMirror relationship, this field is the same on the source side and the destination side. For qtrees in a SnapMirror relationship, the destination side lists the name of the exported snapshot for that qtree on the destination. **Note** A resynchronization may change the name of the base snapshot.
Current Transfer Type	Indicates the kind of transfer now in progress: scheduled, retry, resync, update, initialize, store, or retrieve. This field applies only to the destination side.
Current Transfer Error	Displays an error message if the latest transfer attempt failed.

Additional -l option entries	Description
Contents	Indicates whether the contents of the destination volume or qtree in the active file system are up-to-date replicas or in transition. The field applies only to the destination side. ◆ Under SnapMirror volume replication, the contents are always a replica. ◆ Under SnapMirror qtree replication, the contents are usually a replica, but sometimes are transitioning.
Last Transfer Type	Indicates the kind of transfer previously performed: scheduled, retry, resync, update, initialize, store, or retrieve. This field applies only to the destination side.
Last Transfer Size	Shows the amount of data (in KB) transferred in the last successful transfer.
Last Transfer Duration	Shows the elapsed time for the last successful transfer to complete. If the transfer failed and restarted, this includes time waiting to restart the transfer. If a transfer aborted and was retried from the beginning, it includes only the time required for the final successful attempt.
Last Transfer From	This field applies only to the destination side and shows the name of the source filer and volume or qtree. This field is useful if you have changed the source in the /etc/snapmirror.conf file but the data is actually from the old source. For `snapmirror retrieve` (from tape) operations, this field lists the tape device used in the retrieve operation.

Checking SnapMirror data transfer logs

Why you check data transfer logs

You can use the `options snapmirror.log.enable` command to check SnapMirror data transfer logs. You can find out whether transfers are occurring as planned, how long the transfers take, and how well the system setup works. You find this information in the SnapMirror log file.

What checking data transfer history shows you

The SnapMirror log file shows you

- The start time and the end time of the SnapMirror logging process
- The start time, end time, and size of each transfer
- Any abnormal termination and restart of a transfer
- Other SnapMirror-related activities

What you can do with the information recorded

You can use the raw information provided to do the following:

- Calculate the average transfer size
- Calculate the average transfer time
- Look at the number of successful transfers and the failure rate
- Tune the schedule
- Create a notifier for aborted transfers
- Monitor performance on a per-volume level
- Be assured that things are working as planned

How to find out whether SnapMirror logging is turned on or off

SnapMirror logging is turned on by default. However, if you need to find out whether SnapMirror logging is turned on or off, complete the following step.

Step	Action
1	Enter the following command on the filer for which you want the information: **`options snapmirror.log.enable`** **Result:** SnapMirror reports whether logging is enabled or not.

Example:

```
filerA> options snapmirror.log.enable
snapmirror.log.enable      on
```

How to turn SnapMirror logging on

To turn SnapMirror logging on, complete the following step.

Step	Action
1	Enter the following command on the filer for which you want the log: **`options snapmirror.log.enable on`** **Result:** SnapMirror enables the logging of transfers for the filer. This setting is persistent across reboots.

Where the log files are kept

SnapMirror keeps the current log on the root volume of the executing filer, as /etc/log/snapmirror.0. A new log file is generated every week as /etc./log/snapmirror.0. Older log files are renamed /etc./log/snapmirror.[1-5] and the oldest log file is deleted. The log files can be read with a text editor.

Format of the log files

The log file is in the following format:

`type timestamp` *`source_filer`*`:`*`source_path dest_filer`*`:`*`dest_path`* `event_info`

type can be one of the following: `src, dst, log, cmd.`

type specifies whether the record is for the source side (`src`) or destination side (`dst`) of the transfer. Certain events apply to only one side. The type `log` indicates a record about the logging system itself, for example, `Start_Logging` and `End_Logging`. The type `cmd` indicates a record of user commands, for example, `Release_command` and `Resync_command`.

`timestamp` is expressed in `ctime` format, for example:

`Fri Jul 27 20:41:09 GMT.`

`event_info` includes the following event names:

```
Request ( IP address | transfer type )
Start
Restart (@ num KB)
End (num KB done)
Abort (error_msg)
Defer (reason)
Rollback_start
Rollback_end
Rollback_failed
Start_Logging
End_Logging
Wait_tape
New_tape
Snapmirror_on
Snapmirror_off
Quiesce_start
Quiesce_end
Quiesce_failed
Resume_command
Break_command
Release_command
Abort_command
Resync_command
Migrate_command
```

The `Request` event on the source side includes the IP address of the filer that made the transfer request; the `Request` event on the destination side includes the type of transfer. At the end of each successful transfer, the `End` event also reports the total size of the transfer in KB. Error messages are included with the `Abort` and `Defer` events.

Examples of log files

The following is an example of a log file from the source side.

```
log Fri Jul 27 20:00:01 GMT - - Start_Logging
cmd Fri Jul 27 20:00:20 GMT - - Snapmirror_on
src Fri Jul 27 20:41:09 GMT filer1:vol1 filer2:vol1 Request (10.56.17.133)
src Fri Jul 27 20:41:32 GMT filer1:vol1 filer2:vol1 Abort (Destination not allowed)
src Fri Jul 27 20:45:31 GMT filer1:vol0 filer1:vol1 Request (10.56.17.132)
src Fri Jul 27 20:45:35 GMT filer1:vol0 filer1:vol1 Start
src Fri Jul 27 20:51:40 GMT filer1:vol0 filer1:vol1 End (26200 KB)
src Fri Jul 27 22:41:09 GMT filer1:/vol/vol1/qtA filer2:/vol/vol1/qtB Request
(10.56.17.133)
src Fri Jul 27 22:41:12 GMT filer1:/vol/vol1/qtA filer2:/vol/vol1/qtB Start
src Fri Jul 27 22:41:13 GMT filer1:/vol/vol1/qtA filer2:/vol/vol1/qtB Abort (Non
unicode directory found in source qtree.)
src Fri Jul 27 22:45:53 GMT filer1:/vol/vol1/qtb filer2:/vol/vol1/qsmb Request
(10.56.17.133)
src Fri Jul 27 22:45:56 GMT filer1:/vol/vol1/qtb filer2:/vol/vol1/qsmb Start
src Fri Jul 27 22:45:59 GMT filer1:/vol/vol1/qtb filer2:/vol/vol1/qsmb End (3800 KB)
cmd Fri Jul 27 22:50:29 GMT filer1:/vol/vol1/qtb filer2:/vol/vol1/qsmb Release_command
```

The following is an example of a log file from the destination side.

```
dst Fri Jul 27 22:50:18 GMT filer1:vol0 filer1:vol1 Request (Initialization)
dst Fri Jul 27 22:50:20 GMT filer1:vol0 filer1:vol1 Abort (Destination is not
restricted)
dst Fri Jul 27 22:57:17 GMT filer1:/vol/vol1/qtA filer2:/vol/vol1/qtB Request
(Initialize)
dst Fri Jul 27 22:57:24 GMT filer1:/vol/vol1/qtA filer2:/vol/vol1/qtB Start
dst Fri Jul 27 22:57:36 GMT filer1:/vol/vol1/qtA filer2:/vol/vol1/qtB End (55670 KB)
dst Fri Jul 27 23:10:03 GMT filer1:/vol/vol1/qtA filer2:/vol/vol1/qtB Request
(Scheduled)
dst Fri Jul 27 23:10:07 GMT filer1:/vol/vol1/qtA filer2:/vol/vol1/qtB Start
dst Fri Jul 27 23:10:18 GMT filer1:/vol/vol1/qtA filer2:/vol/vol1/qtB End (12900 KB)
cmd Sat Jul 28 00:05:29 GMT - filer2:/vol/vol1/qtB Quiesce_start
cmd Sat Jul 28 00:05:29 GMT - filer2:/vol/vol1/qtB Quiesce_end
cmd Sat Jul 28 00:05:40 GMT - filer2:/vol/vol1/qtB Break_command
cmd Sat Jul 28 00:41:05 GMT filer1:/vol/vol1/qtA filer2:/vol/vol1/qtB Resync_command
log Sat Jul 28 00:41:10 GMT - - End_Logging
```

The following is an example of a log file from a retrieve (from tape) request.

```
dst Fri Jun 22 03:07:34 GMT milton:rst01 milton:bigtwo Request
(retrieve)
dst Fri Jun 22 03:07:34 GMT milton:rst01 milton:bigtwo Start
dst Fri Jun 22 05:03:45 GMT milton:rst01 milton:bigtwo Wait_tape
dst Fri Jun 22 15:16:44 GMT milton:rst01 milton:bigtwo New_tape
```

```
dst Fri Jun 22 17:13:24 GMT milton:rst01 milton:bigtwo Wait_tape
dst Fri Jun 22 17:56:43 GMT milton:rst01 milton:bigtwo New_tape
dst Fri Jun 22 18:10:37 GMT milton:rst01 milton:bigtwo End
(98602256 KB)
```

How to turn off the SnapMirror log

To turn off the SnapMirror log process, complete the following step.

Step	Action
1	Enter the following command on the filer for which you want to disable the log: **`options snapmirror.log.enable off`** **Result:** SnapMirror disables the logging process.

Aborting a SnapMirror transfer

What aborting a transfer does

You can use the `snapmirror abort` command to abort a volume or qtree copy operation before the source finishes transferring data to the destination. The transfer can be the result of a scheduled update, a manual update, or an initialization. If you abort a copy operation, data transfer stops and SnapMirror is put in a restartable mode. If you abort a transfer that has been aborted before, you cannot restart that transfer again.

If you use the `-h` option (hard abort) with the `snapmirror abort` command, you cannot restart the transfer.

How to tell whether you can restart an aborted transfer

To find out whether you can restart an aborted copy operation, check the output of the `snapmirror status` command. If the status message is "Idle with restart checkpoint (*n X*B done)," the transfer can be restarted.

Example:

```
Source          Destination       State          Lag        Status
filerA:vol0     filerA:vol1       Broken-off     29:09:58   Idle
with restart checkpoint (135 MB done)
```

Note

You can abort only transfers that are displayed by the `snapmirror status` command.

Aborting a transfer

To abort a SnapMirror data transfer, complete the following step.

Step	Action
1	From either the source or the destination filer, enter the following command: **snapmirror abort [-h] {[*dest_filer*:]*dest_volume* | [*dest_filer*:]/vol/*volume_name/qtree_name* ...}** -h specifies a hard abort; the transfer cannot be restarted. SnapMirror stops the transfer and clears the restartable transfer log. This option applies only to the SnapMirror destination. *dest_filer* is the name of the destination filer. *dest_volume* is the destination volume. /vol/*volume_name/qtree_name* is the path name of a destination qtree. If no destination filer is specified, the local host's name is used for the filer name. You can enter more than one destination volume. You can obtain the destination filer and volume from the snapmirror status output. If no destination volume or qtree is specified, the command returns an error message; it does not abort all transfers. To abort all transfers, use the snapmirror off command. If you enter an invalid SnapMirror destination (one that is not shown in the output of the snapmirror status command), the command fails and displays an error message.

Example:
```
filerA> snapmirror abort vol1 filerB:vol2
filerC:/vol3/qtree3

snapmirror abort: Aborting transfer to vol1 filerB:vol2
filerC:/vol3/qtree3
```

Result: SnapMirror aborts the transfer to vol1 on filerA, where the command was entered, and aborts the transfer to vol2 on filerB and the transfer to qtree 3 in vol3 on filerC.

Stabilizing (quiescing) destinations before taking a snapshot

Why to block data transfers to a destination temporarily

You may need to temporarily stop transfers to a destination. For example, if you want to create a snapshot of a SnapMirror destination volume or qtree that contains a database, you may need to ensure that its contents are stable during the snapshot. You can use the `snapmirror quiesce` command to block transfers to the destination after the destination reaches a stable state.

What the quiesce command does

The `snapmirror quiesce` command waits for all existing transfers to both volumes and qtrees to complete and blocks any further updates. If a qtree is not in a stable state (is in transition), the `snapmirror quiesce` command forces it into a stable state.

You can quiesce *only* volumes and qtrees that are online and that are SnapMirror destinations.You cannot quiesce a restricted or offline volume or a qtree in a restricted or offline volume.

The `snapmirror quiesce` command stops a volume or qtree from acting as a SnapMirror destination, but does not prevent it from acting as a SnapMirror source.

Note

The quiesced state persists across reboots.

How to quiesce data transfers

To stabilize and halt updates to data on a SnapMirror destination volume or qtree, complete the following step.

Step	Action
1	Enter the following command on the filer on which you want to block transfers: **`snapmirror quiesce {`*`dest_volume`*` | /vol/`*`volume_name`*`/`*`qtree_name`*`}`** *dest_volume* is the name of the destination volume. *qtree_name* is the name of a qtree in *volume_name.*

Example 1: `filerA> snapmirror quiesce vol1`

```
snapmirror quiesce: in progress.
snapmirror quiesce: vol1: successfully quiesced
```

Result: SnapMirror stops any further data transfers to vol1.

Example 2: `filerA> snapmirror quiesce vol2`

```
snapmirror quiesce: in progress.
This can be a long-running operation. Use Control-C to interrupt.
.....................................
snapmirror quiesce: vol2: successfully quiesced
```

Result: SnapMirror waits for a transfer to finish and stops any further data transfers to vol2.

Example 3: `filerA> snapmirror quiesce /vol/vol1/qtree1`

Result: SnapMirror stops data transfers to qtree1 in vol1.

If you use the `snapmirror break` command on a destination that is quiesced, the quiesce condition is automatically cleared when the destination becomes writable.

Resuming transfers after quiescing a destination

You can use the `snapmirror resume` command to restore the capability for data transfer to a volume or qtree you have quiesced.

To resume SnapMirror transfers, complete the following step.

Step	Action

1	Enter the following command for the filer on which you want to resume transfers:
	snapmirror resume {*dest_volume* | /vol/*volume_name*/*qtree_name*}
	dest_volume is the name of the destination volume.
	qtree_name is the name of a qtree in *volume_name.*
	Example:
	`filerA> snapmirror resume vol2`
	`snapmirror resume: vol2: Successfully resumed`
	Result: SnapMirror resumes normal data transfer capability for vol2.

Turning off or changing scheduled updates for volumes or qtrees

Why you turn off scheduled updates

You can edit the destination's etc/snapmirror.conf file to change or turn off scheduled updates for a particular volume or qtree if you decide that there is no need to update the destination. For example, you might want to change the time or frequency of scheduled updates if the pattern of use or the configuration of the filers has changed. Or, if you want to use the volume for a different purpose, you can change the destination to a writable volume.

Note

Editing entries in the destination's etc/snapmirror.conf file to turn off scheduled updates does not change the destination to a writable volume. If you want to change the destination to a writable volume or qtree, you use the `snapmirror break` command to turn the destination into a writable volume or qtree and the `snapmirror release` command to allow SnapMirror to delete the snapshots it no longer needs on the source. See "Converting a destination to a writable volume or qtree," on page 139 and "How to release a source from a SnapMirror relationship," on page 142.

When you can turn off or change scheduled updates

You can edit the destination's etc/snapmirror.conf file to turn off or change scheduled updates at any time, even when data transfer is underway. The destination remains the same as before the transfer. The snapshot taken in the source for the data transfer remains, but it can be deleted and replaced by a new snapshot the next time the destination is updated.

Changing scheduled updates for one volume or qtree

To change scheduled updates for one volume or qtree, complete the following step.

Step	Action

1	In the destination's /etc/snapmirror.conf file, edit the destination volume or schedule information to specify the configuration you want.
	Example:
	Original update schedule:
	`filerA:vol0 filerA:vol1 - 0 23 * 1,3,5`
	`filerA:vol1 filerB:vol6 - 0 23 * 1,3,5`
	Changed update schedule:
	`filerA:vol0 filerA:vol2 - 0 23 * 1,3,5`
	`filerA:vol1 filerB:vol6 - 0 23 * 2,4,6`

Turning off scheduled updates for one volume

To turn off scheduled updates for one volume or qtree, complete the following step.

Step	Action

1	Either delete the entry in the /etc/snapmirror.conf file or change the entry by
	◆ Commenting out the entry by preceding it with a pound sign (#)
	Example:
	`filerA:vol0 filerA:vol1 - 0 23 * 1,3,5`
	`filerA:vol1 filerB:vol6 - 0 23 * 1,3,5`
	`#filerB:vol1 filerC:vol2 - 0 23 * 1,3,5`
	◆ Putting a dash (-) in one of the schedule fields (minute/hour/dayofmonth/dayofweek)
	Example:
	`filerA:vol0 filerA:vol1 - 0 23 * 1,3,5`
	`filerA:vol1 filerB:vol6 - 0 23 * 1,3,5`
	`filerB:vol1 filerC:vol2 - - 23 * 1,3,5`

Note

Deleting or commenting out a destination or putting a dash in one of the schedule fields of a destination in the /etc/snapmirror.conf file does not prevent you from performing manual updates to that destination.

Turning off SnapMirror updates for the filer

Why you turn off updates

You can use the `snapmirror off` command to turn off updates, both scheduled and manual, for the entire filer at any time, even when copying is underway. Any active transfer is aborted when you turn off SnapMirror for the filer. The destination remains unchanged after you turn off updates.

This process affects all SnapMirror transfers for the filer, whether the filer is the source or the destination of the SnapMirror relationship.

To turn off scheduled and manual updates for the entire filer, complete the following steps.

Step	Action
1	Enter the following command on both the source filer and destination filer to disable SnapMirror: **`options snapmirror.enable off`** Alternatively, you can still use the older command to turn SnapMirror off: **`snapmirror off`** **Result:** If SnapMirror is currently transferring data from one volume or qtree to another, the transfer aborts immediately. The destination remains the same as before the transfer. The snapshot taken in the source volume for the data transfer remains. SnapMirror stops monitoring the /etc/snapmirror.conf file for changes. Entering the `snapmirror off` command on the destination filer alone does not affect SnapMirror on the source filer. Other filers can continue to copy data from the source filer. **Note** Both the `snapmirror off` command and the `snapmirror.enable off` option are persistent across reboots.

Step	Action
2	If the `snapmirror on` command is in the /etc/rc file, remove the command (to keep the current setting after reboot). Otherwise the setting in the /etc/rc file overrides the command you entered.

Converting a destination to a writable volume or qtree

Why to convert a destination to a writable volume

You can use the `snapmirror break` command to convert a SnapMirror destination, with read-only status, to a writable volume or qtree. You might want to convert a destination to a writable volume or qtree to perform one of the following tasks:

- Data migration—You want to move your user's working data from one volume or qtree (your current source volume or qtree) to another volume or qtree (your current destination volume or qtree) and make that data accessible and writable in its new location.
- Disaster recovery—In case your source volume or qtree is suddenly unavailable, you want your current destination volume or qtree to serve as the substitute for your users' retrieval and input source.
- Application testing—You want to make your current destination volume or qtree writable to test a new application on a mirrored replication of your current data rather than risk corruption of original data on the source volume or qtree.

Converting the destination to a writable volume or qtree lets you use data on the destination for these situations or in any other situation in which the original source is unavailable.

Preserving quota restrictions

Quotas are always disabled on a SnapMirror volume destination, regardless of whether quotas are enabled on the source volume. If you try to enable quotas on a volume destination, SnapMirror displays an error message. Quotas are not disabled on SnapMirror destination qtrees.

If the source volume or qtree and the destination reside on different filers, and you want the same quota restrictions to be applied after you make the destination writable, the destination filer must have an /etc/quotas file that includes all the entries from the /etc/quotas file used by the source filer.

- If you use SnapMirror replication for data migration, you can copy the /etc/quotas entries from the source filer to the /etc/quotas file of the destination filer before you use the `snapmirror break` command to make the destination writable.
- If you use SnapMirror replication for backup and potential disaster recovery, you must keep a copy on the destination filer of all /etc/quotas entries used

by the source filer at all times. That way, you can apply the quota entries to the destination volume or qtree if the source filer becomes unavailable.

Converting to a writable volume or qtree

To convert a destination to a writable volume or qtree, complete the following steps.

Step	Action
1	On the destination filer use the `snapmirror break` command to make the destination volume or qtree writable. ◆ To make a destination volume writable, enter the following command on the destination filer. **snapmirror break *volume_name*** ◆ To make a destination qtree writable, enter the following commands on the destination filer. **snapmirror quiesce /vol/*volume_name*/*qtree_name*** **snapmirror break /vol/*volume_name*/*qtree_name***
2	If you want to enable quotas on the former destination volume, carry out the following steps. **1.** Edit the /etc/quotas file on the former destination filer so that, after the conversion, the former destination includes the same quota restrictions as the source volume. If the original source volume uses per-volume quotas, replace the original source volume name with the former destination name in the quota entries. **2.** Enter the following command to enable quotas on the former destination: **quota on *volume_name***

Step	Action
3	Consider the following optional measures: ◆ If you want to stop a SnapMirror source from trying to update a broken-off destination, you can delete or comment out the entry in the /etc/snapmirror. conf file. Otherwise SnapMirror continues to try to update the destination. ◆ You might also want to use the `options fs_size_fixed off` command to turn off the option that restricts the size of the file system on a destination volume. **Note** However, after you set `options fs_size_fixed off`, the ability of the destination and source volumes to resync is not guaranteed. For more information, see "How snapmirror initialize matches source and destination volume size," on page 108.

After using snapmirror break

After using the `snapmirror break` command to temporarily break a SnapMirror relationship between a source and destination, you can use other SnapMirror commands to either make the break permanent or restore or redefine the SnapMirror relationship.

- Use the `snapmirror release` command to make the break permanent. See "Releasing partners from a SnapMirror relationship," on page 142.
- Use the `snapmirror resync` command to restore or redefine the SnapMirror relationship. See "Resynchronizing SnapMirror," on page 144.

Releasing partners from a SnapMirror relationship

When to release partners from a SnapMirror relationship

If you want to permanently end a SnapMirror relationship between a source and destination volume or qtree, you invoke different commands on the source and destination filers.

- On the source filer, you use the `snapmirror release` command. Releasing a source from a destination volume or qtree allows the source to delete its base snapshot for the SnapMirror relationship. See "How to release a source from a SnapMirror relationship" below.
- On the destination filer, you use the `snapmirror break` command, and carry out some additional operations. See "How to scrub a destination of a broken SnapMirror relationship," on page 143.

How to release a source from a SnapMirror relationship

To release a source from a SnapMirror relationship, complete the following step.

Step	Action
1	On the source filer, enter the following command: **`snapmirror release {`*`source_volume`*` | `*`qtree_path`*`} [`*`dest_filer`*`:]{`*`dest_volume`*` | `*`qtree_path`*`}`** *source_volume* or *qtree_path* is the name of the source volume or path to the qtree that you want to release from the destination. *dest_filer* is the name of the filer where the destination is located. *dest_volume* or *qtree_path* is the name of the volume or path to the qtree that is the destination. If you do not enter the name of the destination filer, SnapMirror uses the name of the filer on which you entered the command. **Result:** SnapMirror frees all resources on the source filer that had been dedicated to the SnapMirror relationship.

Volume example: `filerA> snapmirror release vol0 filerB:vol2`

Qtree example: `filerA> snapmirror release vol/vol1/qtree2 filerB:/vol/vol2/qtree5`

How to scrub a destination of a broken SnapMirror relationship

In order to permanently break a SnapMirror relationship on a destination volume or qtree, you need to carry out extra steps in addition to invoking `snapmirror break` on the destination filer and `snapmirror release` on the source filer. Unless these extra steps are taken, the snapshots associated with the broken relationship remain stored on the destination filer, and a `snapmirror status` command will continue to list the former destination object as a current destination object. To scrub a destination of a SnapMirror relationship, complete the following steps.

Step	Action
1	If you have not already done so, ◆ invoke the `snapmirror break` command on the destination filer to break the SnapMirror relationship between the source and destination objects. For details see "Converting to a writable volume or qtree," on page 140. ◆ invoke the `snapmirror release` command on the source filer to release the source object from this SnapMirror relationship. For details see "How to release a source from a SnapMirror relationship," on page 142.
2	On the destination filer, invoke the `snapmirror status -l` command to determine which snapshot basename is associated with the SnapMirror relationship that you just broke. ◆ For broken SnapMirror volume relationships enter: `snapmirror status -l` *`dest_vol`* ◆ For broken SnapMirror qtree relationships enter: `snapmirror status -l /vol/`*`dest_vol`*`/`*`dest_qtree`* In the detailed output that is displayed, note the snapshot basename associated with the SnapMirror relationship that you just broke.
3	On the destination filer, delete the snapshot set that you displayed in Step 2. Enter: `snap delete` *`snapshot_basename`*
4	Through the Adminhost client, edit the /etc/snapmirror.conf file on the destination filer. Locate and delete the entry that specifies the SnapMirror relationship you want to end. For details on editing the /etc/snapmirror.conf file, see "snapmirror.conf file entry syntax," on page 94.

Resynchronizing SnapMirror

Why to resynchronize a source and destination volume

You can use the `snapmirror resync` command to restore or redefine a SnapMirror source or destination relationship that was broken with the `snapmirror break` command. You might want to resynchronize a source and a destination volume or qtree when

- You are changing the current source to a different volume or qtree
- You make a destination volume writable for application testing and then want to make it a SnapMirror destination again
- You need to recover from a disaster that disabled the source
- You want to reverse the functions of the source and the destination

What the snapmirror resync command does

After the `snapmirror break` command, you apply the `snapmirror resync` command to either the original SnapMirror destination or the original source.

- Applied to the original destination— the `snapmirror resync` command will put a volume or qtree back into a SnapMirror relationship and resynchronize its contents with the source without repeating the initial transfer.
- Applied to the source volume— the `snapmirror resync` command can turn the source volume into a copy of the original destination volume. In this way, the roles of source and destination can be reversed.

How the snapmirror resync command helps minimize data loss

The `snapmirror resync` command enables you to re-establish a broken SnapMirror relationship without a lengthy baseline transfer.

This command offers the choice of either source or destination to serve as the source in the restarted SnapMirror relationship. It finds the newest common snapshot (NCS) shared by the two volumes or qtrees, and removes all newer information on the filer on which the command is run.

Note

The `snapmirror resync` command requires that the two volumes or qtrees have at least one snapshot in common. You can resynchronize a volume or qtree to any other volume or qtree as long as both have at least one snapshot in common.

Resynchronization will cause the loss of all data written to the destination after the base snapshot was made. The `snapmirror resync` command informs you what data could be lost during the resynchronization and requests permission to proceed. If you want to save the data on the destination, you can stop the resynchronization, manually copy the desired data elsewhere, and reissue the `snapmirror resync` command to the destination.

Resynchronizing a SnapMirror relationship

To resynchronize a destination to a source, complete the following steps.

Step	Action
1	From the destination filer, enter the following command: **`snapmirror resync [`*`options`*`]`** **`[`*`dest_filer`*`:]{`*`dest_volume`*` | /vol/`*`qtree_path`*`}`** *options* can be any of the following: ◆ `-n` does not execute the resynchronization, but displays what would be done if the `snapmirror resync` command were run. You can use this option to find whether you have a snapshot on the source and on the destination that can be used as the newest common snapshot (base snapshot) so that you can resync a specific SnapMirror relationship. ◆ `-f` forces the operation to proceed without prompting you for confirmation. ◆ `-k` *n* sets the maximum transfer speed to *n* kilobytes per second. This option has the same effect as the `kbs` argument in the /etc/snapmirror.conf file. ◆ `-S` [*source_filer*:]{*source_volume* | *qtree_path*}specifies the filer and volume or qtree you want to use as the source for resynchronization. The source specified by the `-S` option must match a source entry in the /etc/snapmirror.conf file. If entries exist but the source does not match, the operation displays an error message and terminates. If there is no entry for the specified source, the command runs. If the `-S` option is not set, the source must be specified in the /etc/snapmirror.conf file. If it is not specified, the operation displays an error message and terminates.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td></td><td colspan="2">dest_filer is the name of the destination filer.

dest_volume or <code>/vol/qtree_path</code> is the destination volume or qtree. If it is a scheduled destination as specified in the /etc/snapmirror.conf file, that source volume or qtree is assumed to be the source. If the destination volume or qtree specified is not in a scheduled relationship, then the <code>-S</code> option must be used to provide a source.

Result: SnapMirror identifies the newest common snapshot, which will be used as the base for resynchronization, and generates a list of snapshots on the destination volume that
◆ Are newer than the base snapshot and will be deleted
◆ Are older than the base snapshot and have already been deleted from the source

Note
For qtree resynchronization, only the common snapshot is displayed.

SnapMirror then prompts you to choose whether to continue.</td></tr>
<tr><td>2</td><td>If...</td><td>Then...</td></tr>
<tr><td></td><td>You want to
◆ Reestablish the SnapMirror pair
◆ Delete the listed snapshots on the destination volume (if you are resynchronizing volumes)</td><td>Type <code>y</code> at the prompt.

Result: SnapMirror
◆ Deletes the listed snapshots on the destination volume (if you are resynchronizing volumes).
◆ Makes the destination read-only.
◆ Initiates an update of the destination.</td></tr>
</table>

Step	Action	
	If...	**Then...**
	You do not want to lose the data in a snapshot that was created after the common snapshot on the destination, but you want to resynchronize the two volumes or qtrees after the data is saved	◆ Type `n` at the prompt. ◆ Manually copy the data you want to save to the source or other volume. ◆ Return to Step 1 to rerun the `snapmirror resync` command.
	You do not want to reestablish the SnapMirror relationship	Type `n` at the prompt. **Result:** SnapMirror terminates the command.

Volume example: `filerB> snapmirror resync filerB:vol2`

```
The resync base snapshot will be vol2(0001234567)_d.4
These newer snapshots will be deleted from the destination:
hourly.0
hourly.1
These older snapshots have already been deleted from the source and
will be deleted from the destination:
vol2(0001234567)_d.3
Are you sure you want to resync the volume?
```

Qtree example: `filerB> snapmirror resync -S filerA:/vol/vol2/qtreeBob filerB:/vol/vol3/qtreeBak`

```
The resync base snapshot will be vol2(0001234567)_d.4
Data could be lost as a result of this operation.
Are you sure you want to resync the volume?
```

If no common snapshot on the source and destination exists

If SnapMirror cannot find a common snapshot on the source and destination to use as the basis for resynchronization, resynchronization is not possible. SnapMirror generates an error message that states the problem and terminates the command. You must re-initialize the destination to establish the SnapMirror relationship.

Database application testing: a special use for snapmirror resync

Testing software applications that run on a database can sometimes change or even corrupt the database. To ensure that you do not lose data while testing such software applications, you can copy the data to another volume for testing purposes, break the destination volume to return it to writable state, and run the test application on it. Upon completion of the test, you can resynchronize the source and the destination volume to restore the data and repeat the testing as often as you need.

In the following procedure, you can use a combination of the `snapmirror break` and `snapmirror resync` commands to

- make a destination volume writable for testing.
- restoring the newly writable volume to its original state if further testing is required.

To set up a destination for testing software applications, complete the following steps.

Step	Action	
1	Create or choose a volume or qtree to be used as a destination for the volume or qtree containing the database. (This example uses a qtree called Testqtree.)	
2	Follow the steps described in “Enabling SnapMirror,” on page 100.	
3	On the destination filer, enter the following command to make the destination writable: **`snapmirror break Testqtree`**	
4	Run the application on the data in the former destination (Testqtree).	
5	Check the data in the former destination (Testqtree).	
6	**If the data...**	**Then...**
	Has been altered in some way that is not useful and you want to import a fresh copy of the data for further testing	From the destination filer, enter the following command: **`snapmirror resync Testqtree`** **Result:** SnapMirror makes the former destination volume into a SnapMirror destination again and updates the destination with the latest data.

Step	Action	
	Has not been altered deleteriously, or you wish to stop testing	You have finished.
7	Repeat Steps 4, 5, and 6 until you are satisfied with the testing.	

Disaster recovery: a special use for snapmirror resync

When disaster disables the source of a SnapMirror relationship, you can use the `snapmirror resync` command as part of a strategy to update the repaired source and reestablish the original configuration of the filers.

Example summary: In the following example, the original source (the one disabled by the disaster) is FilerA:vol/volA and the original destination is FilerB:/vol/volB. You use a combination of `snapmirror break` and `snapmirror resync` or `snapmirror initialize` commands:

- to temporarily make FilerB:volB the source and FilerA:volA the destination to restore mirrored data back to FilerA:volA and to update FilerA:volA.
- to restore FilerA:/vol/volA and FilerB:volB to their original roles as SnapMirror source and SnapMirror destination volume.

What data is preserved: In this example, all data from the last scheduled SnapMirror snapshot before the source was disabled and all the data written to FilerB:vol/volB after it was made writable is preserved. Any data written to FilerA:vol/volA between the last SnapMirror snapshot and the time that FilerA:vol/volA was disabled is not preserved.

Step	Action
1	After the source volume (in this case, FilerA:volA) is disabled, the administrator uses the `snapmirror break` command to make the destination volume, FilerB:volB, writable. **`snapmirror break FilerB:volB`**
2	The administrator redirects the clients of source FilerA to source FilerB. The former clients of FilerA are now accessing and writing to FilerB.

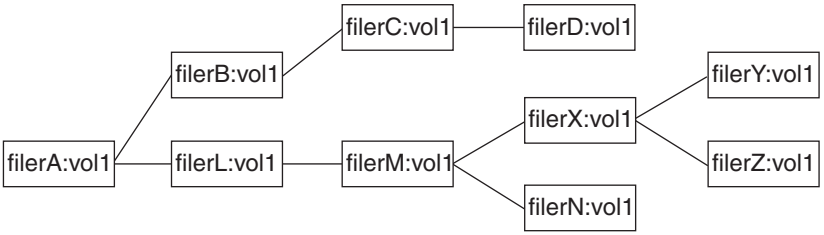
Step	Action
3	The administrator temporarily makes the original source volume into a read-only destination volume. ◆ If FilerA:volA is recoverable, and its data is intact, then the administrator on FilerA uses the `snapmirror resync` command to resynchronize FilerA with FilberB. **`snapmirror resync -S FilerB:VolB FilerA:volA`** ◆ If FilerA:volA is unrecoverable, the administrator makes a new volA on FilerA, and from FilerA, initializes FilerA:volA from FilerB. **`snapmirror initialize -S FilerB:volB FilerA:volA`** This command also makes FilerA:volA a read-only destination.
4	The administrator then redirects the clients from FilerB to FilerA. The clients cannot access or write to FilerA:volA, but they are no longer writing new data to FilerB:volB.
5	The administrator updates FilerA:volA from FilerB to transfer the latest data from Filer B. **`snapmirror update -S FilerB:volB FilerA:volA`**
6	Now, the administrator uses the `snapmirror break` command to make FilerA:volA writable. In FilerA the administrator enters: **`snapmirror break volA`**
7	From FilerB, the administrator uses the `snapmirror resync` command to make FilerB, the original destination, the destination again. **`snapmirror resync volB`**

Copying from one destination to another in a series (cascading)

When to copy from a destination

It can be useful to copy from a SnapMirror destination when the data you want to copy is on a destination that is closer to you than the same data on a source. It is also useful to copy from a destination when you need to copy data from one site to many sites. Instead of propagating the data from one central master site to each of the destination sites, which would require expensive network connections or excessive CPU time, you can propagate the data from one destination to another destination and from that one to the next, in a series.

In the example below, the data on filerA is copied on nine different filers, but only two of the filers copy the data directly from the source. The other seven filers copy the data from a destination site.



Note

The cascading procedure is supported for SnapMirror volume replication only. Cascading is not supported for SnapMirror qtree replication.

How to copy from a destination

You can copy from a destination volume the same way you copy from a writable source volume. For information on how to set up a destination, see "Enabling SnapMirror," on page 100.

Sample cascade setup

To support a series of cascading volume destinations as shown in the diagram above, the /etc/snapmirror.conf file in each of the cascaded filers would look like this:

```
filerA:vol1 filerB:vol1 - 15 * * 1,2,3,4,5
filerA:vol1 filerL:vol1 - 15 * * 1,2,3,4,5
filerB:vol1 filerC:vol1 - 25 * * 1,2,3,4,5
filerC:vol1 filerD:vol1 - 35 * * 1,2,3,4,5
```

```
filerL:vol1 filerM:vol1 - 25 * * 1,2,3,4,5
filerM:vol1 filerX:vol1 - 35 * * 1,2,3,4,5
filerM:vol1:filerN:vol1 - 35 * * 1,2,3,4,5
filerX:vol1 filerY:vol1 - 45 * * 1,2,3,4,5
filerX:vol1 filerZ:vol1 - 45 * * 1,2,3,4,5
```

Note

When specifying the destination update schedule in the snapmirror.conf file, stagger the update times instead of starting multiple destination updates at the same time. If SnapMirror does not have enough resources to perform all scheduled destination updates, it postpones some updates until it has enough resources to do so. As a result, SnapMirror might need to perform subsequent updates at times that are different from those you specify in the snapmirror.conf file.

How SnapMirror handles snapshots for cascading destinations

SnapMirror retains the snapshots on the original source volume needed for transfers to destinations further down the line. Snapshots that are still needed by a destination are labeled "snapmirror" in the output of the `snap list` command. SnapMirror deletes the snapshots it no longer needs.

If you remove a destination from the cascade, you use the `snapmirror release` command from the immediate source to tell SnapMirror to delete the snapshots associated with that destination.

Listing SnapMirror destinations for a volume in a cascading series

To display the SnapMirror destinations for a source volume, complete the following step.

Step	Action
1	From the filer with the volume serving as the source, enter the following command: **`snapmirror destinations [-s] [`*`volume_name`*`]`** The `-s` option generates a list of the names of the snapshots retained for each destination. *volume_name* is the name of the source volume for which you want to see the destinations.

Sample snapmirror destinations command output

Following is an example of the `snapmirror destinations` command output for the configuration in the diagram above.

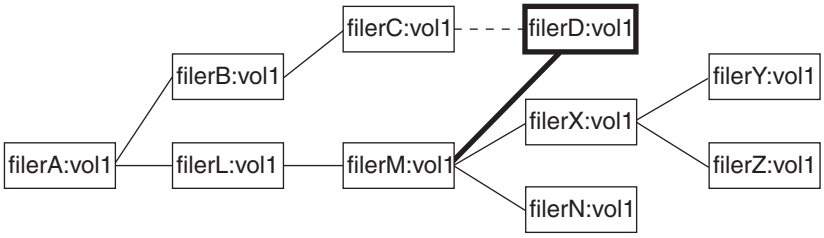
```
filerA> snapmirror destinations vol1

Path        Destination
/vol/vol1   filerB:vol1->filerC:vol1->filerD:vol8
/vol/vol1   filerL:vol1->filerM:vol1->filerX:vol1->filerY:vol1
/vol/vol1   filerL:vol1->filerM:vol1->filerX:vol1->filerZ:vol1
/vol/vol1   filerL:vol1->filerM:vol1->filerN:vol1
```

If you do not specify a volume name in the command, the output includes information about each destination volume on the filer.

Restructuring a cascade

You might restructure a cascade to balance loading of your filers; to use a filer or volume for a different purpose; or to perform upgrades, maintenance, or repairs. For example, in the cascade structure below, you might want to make filerD:vol1 a destination of filerM:vol1 instead of a destination of filerC:vol1.



To restructure the relationship of the destinations in a cascade, complete the following steps.

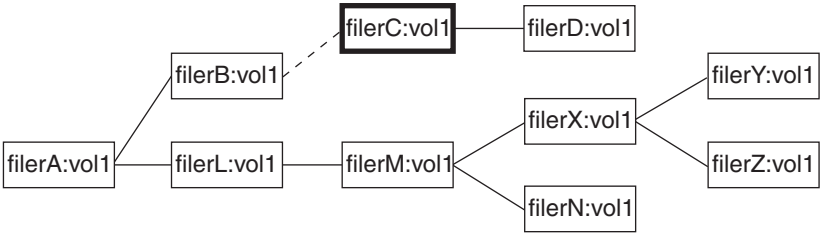
Step	Action
1	On the destination filer, change the /etc/snapmirror.conf file to indicate the new source for the destination. **Example:** `filerM:vol1 filerD:vol1 - 35 * * 1,2,3,4,5`

Step	Action	
2	**If...**	**Then**
	The newest snapshot on the destination exists on the source	Use the following command to update the destination from the new source: **`snapmirror update -S source_volume dest_filer:dest_volume`** **Example:** `snapmirror update -S filerM:vol1 filerD:vol1`
3	**If...**	**Then**
	The newest snapshot on the destination does not exist on the source	Perform one of the following tasks: **1.** Update the new source from the original source using the `snapmirror update` command. Wait for the destination to update. **2.** Make the destination writable using the `snapmirror break` command. Then resynchronize the destination with the new source using the `snapmirror resync` command.
4	Release the former source using the following command: **`snapmirror release source_volume [[dest_filer:]dest_volume]`** **Example:** `filerC> snapmirror release filerC:vol1 filerD:vol1`	

Example of disconnecting a destination from a cascading series:

For the example shown in the diagram below, suppose that from filerB you enter the following command:

```
snapmirror release vol1 filerC:vol1
```



These results follow:

- filerA:vol1 continues to be the source for the destination filerB:vol1.
- filerC:vol1 no longer copies from filerB:vol1. SnapMirror retains snapshots for filerC and below.
- If filerC requests an update from filerB, the destination is reestablished if it is still not writable and the base snapshot still exists on the source.
- filerD:vol1 still copies filerC:vol1.
- All the destinations that depend on filerL:vol1 continue functioning as before.

You can check that the destination was released by running the `snapmirror destinations` command on filerA, as follows:

```
filerA> snapmirror destinations -s filerA:vol1
Volume Snapshot                      Destination
vol1   filerB(0015269532)_vol1.37   filerB:vol1
vol1   filerL(0015269532)_vol1.42   filerL:vol1->filerM:vol1-
>filerXvol1->filerY:vol1
vol1   filerL(0015269532)_vol1.42   filerL:vol1->filerM:vol1-
>filerXvol1->filerZ:vol1
vol1   filerL(0015269532)_vol1.42   filerL:vol1->filerM:vol1-
>filerN:vol1
```

Note

If you want the release of a destination to be permanent, you need to either delete the entry in the /etc/snapmirror.conf file or comment out the entry by preceding it with a pound sign (#). Otherwise, SnapMirror continues to try to update the destination.

SnapMirroring a volume to local tape

Why use SnapMirror to local tape

You might want to snapmirror a volume from a source filer to local tape for the following reasons:

- Network transfer time of the baseline transfer between a SnapMirror source and a SnapMirror destination is prohibitive. For details see the following section, "SnapMirror source-to-tape-to-destination scenario," on page 156.
- You are backing up SnapVault secondary storage data to tape for offline storage or to protect against the possible loss of the SnapVault secondary. For details see "SnapVault-to-tape backup scenario," on page 162.

Note

The `snapmirror store` command does not support SnapMirror qtree replication to tape. If you specify a qtree path as the source or destination, SnapMirror returns an error message.

SnapMirror source-to-tape-to-destination scenario

In this scenario, you want to establish a SnapMirror relationship between a source filer and a destination filer over a low-bandwidth connection. Incremental snapshot mirroring from the source to the destination over the low-bandwidth connection is feasible, but the initial base snapshot mirroring is not. In such a case it might be faster to first transfer the initial base snapshot image from source to destination via tape, and then set up incremental SnapMirror updates to the destination filer via the low-bandwidth connection.

Prerequisites: This scenario assumes the following configuration:

- A low-bandwidth connection between the source and destination filers
- A local tape drive attached to the source filer
- A local tape drive attached to the destination filer

Caution

To prevent extended tape-to-filer transfer time, it is recommended that the destination filer disks be the same size and in the same RAID configuration as the source filer disks.

Sequence of SnapMirror operations: You follow this sequence to set up this arrangement:

1. On the source filer, use the `snapmirror store` command to copy all volume snapshots, including the base snapshot, to tape, and use the `snapmirror use` command to continue the copy if more than one backup tape is necessary.

   For more information, see "Copying source-to-intermediate tape," on page 158.

2. Physically transport the backup tapes from the source filer to the destination filer.

3. On the destination filer, use the `vol create` and `vol restrict` commands to set up a SnapMirror target volume.

4. Use the `snapmirror retrieve` command to copy the initial SnapMirror tape to the destination filer and, if necessary, use the `snapmirror use` command to continue the copy if it is stored on more than one backup tape.

   Then, either use the `snapmirror update` command to manually mirror an incremental update from the source to the destination filer over the low-bandwidth connection, or edit the snapmirror.conf file to set up an incremental update schedule from the source to destination filer.

   For more information, see "Initializing a SnapMirror destination via local tape," on page 160.

5. After completing manual or scheduled incremental update over a connection, you can use the `snapmirror release` command to eliminate the source-to-tape relationship and associated snapshot. For more information see "Releasing a SnapMirror source-to-tape relationship," on page 161.

Copying source-to-intermediate tape: To copy a source volume to an intermediate tape, complete the following steps.

Step	Action
1	If you do not know whether the disk geometries (that is, the size and number of disks) of the source volume and the ultimate destination volume match, use the `snapmirror retrieve -g` command to determine if this is so. In the filer consoles of both the source and destination, enter the following command: **`snapmirror retrieve -g vol_name`** For both the source and the destination volume, the system displays the number and block size of the disks it includes. For example, a return value of `7200000x10 7000x10` means that the specified volume consists of 10 disks of 720,000 blocks and 10 disks of 7,000 blocks. ◆ The most desirable result is for the source and destination volume disk geometries to match. ◆ If the geometries do not match, record the disk geometry of the destination volume for later use.
2	At the source filer, load the tape into a local attached tape device.

Step	Action
3	At the source filer, start the data transfer to tape by entering the following command: **snapmirror store [-g *dest_disk_geom*] *source_volume* *dest_tapedevices*** `-g` *dest_disk_geom* applies if the disk geometry of the destination volume, as determined in Step 1, is different from the disk geometry of the source volume. If they are different, use the `-g` parameter to specify the destination volume disk geometry, for example: **-g 140000x10,7000x10** *source_volume* is the volume you are copying. *dest_tapedevices* is the tape device or a comma-separated list of tape devices to which you are copying the volume. **Result:** SnapMirror starts transferring data to tape. This command creates a baseline transfer. If you have to use more than one tape, SnapMirror prompts you for another tape. **Example:** **snapmirror store -g 14000x10,7000X10 vol2 nrst0a,rst1a**
4	If SnapMirror prompts you for another tape, add another tape to the drive, and continue transfer of data to tape by entering the following command: **snapmirror use *dest_tapedevices tape_drive*** *dest_tapedevices* is the tape device or a comma-separated list of tape devices to which you are copying the volume. *tape_drive* is the drive holding the new tape you supplied. **Note** The `snapmirror use` command does not support copying qtrees from tape. If you specify a qtree path as the source or destination, SnapMirror returns an error message.
5	Repeat Step 4 until SnapMirror finishes copying the volume to tape.

Step	Action
6	Follow the procedure described in "Initializing a SnapMirror destination via local tape," on page 160.

Initializing a SnapMirror destination via local tape: To initialize a SnapMirror destination volume via local tape, complete the following steps.

Step	Action
1	Create a volume on the SnapMirror destination filer. See the *System Administrator's Guide* for information about how to create a volume.
2	Put the volume in the restricted state. See the *System Administrator's Guide* for information about how to restrict a volume.
3	Load the tape (made with `snapmirror store`) into the destination filer's local tape device.
4	Start the initialization by entering the following command on the destination filer: **`snapmirror retrieve dest_volume tape_drive`** *dest_volume* is the volume that you are initializing. *tape_drive* is a tape device or a comma-separated list of devices from which you are restoring the volume. **Result:** SnapMirror starts transferring data from the tape. If data is stored on more than one tape, SnapMirror prompts you for the next tape.
5	If Data ONTAP prompts you for another tape, add the next tape to the drive and restart the initialization by entering the following command: **`snapmirror use volume tape_list`** *volume* is the volume to which you are restoring. *tape_list* is the tape device from which you are restoring the volume.
6	Repeat Step 5 until SnapMirror finishes initializing the volume from the tape.

Step	Action
7	If you need to, you can update the data online manually with the following command: **`snapmirror update [-k n] -S source_filer:source_volume [dest_filer:]dest_volume`** `-k` *n* sets the maximum transfer speed to *n* kilobytes per second. This option has the same effect as the `kbs` argument in the /etc/snapmirror.conf file. `-S` *source_filer*:*source_volume* specifies the source filer and volume for the migration. *source_volume* is the volume you want to copy. *dest_filer* is the name of the destination filer. *dest_volume* is the destination volume. For more information about the `snapmirror update` command, see "Updating a destination manually," on page 110. **Note** Alternatively, you can update the baseline transfer automatically with the schedule you set in the /etc/snapmirror.conf. file. See "Creating and editing the snapmirror.conf file," on page 92.

Releasing a SnapMirror source-to-tape relationship: After a successful update of the destination filer over a connection, you no longer require the source-to-tape SnapMirror relationship and its associated snapshot that you established with the `snapmirror store` command. To end the source-to-tape backup and delete the snapshot off the source filer, complete the following steps.

Step	Action

1	In the source filer console, enter the following command:
	`snapmirror status`
	Your system will display at least two SnapMirror relationships:
	◆ the source-to-tape relationship, established when you used the `snapmirror store` command
	◆ the source-to-destination relationship, established when you used the `snapmirror update` command
	Example:
	`>snapmirror status`
	`....`
	`source destination state ....`
	`s_filer:vol1 snapmirror_tape_01_15_03_20:05:32 ...`
	`s_filer:vol1 d_filer:vol1 snapmirrored ...`

2	Release the source-to-tape SnapMirror relationship. For this particular operation, the `snapmirror release` syntax is unique: `snapmirror release` *`source_vol tape_snapshot_name`* **Example:** **`snapmirror release vol1 snapmiror_tape_01_15_03_20:05:32`** **Note** Do not release any other SnapMirror relationship. Those relationships are necessary to continue your incremental updates over the low-bandwidth connection.

SnapVault-to-tape backup scenario

In this scenario, you snapmirror snapshots from a SnapVault secondary storage system to an attached local tape drive for the purposes of backup. In case of data loss on the SnapVault secondary system, you can restore a specified snapshot back to the SnapVault secondary system.

Note

This scenario applies only to backup of SnapVault secondary storage. Do not apply this configuration for any other purpose.

Prerequisites: You need a SnapVault secondary system with an attached local tape drive. The volume on which the secondary storage qtrees reside must also be configured as a SnapMirror source volume.

Sequence of SnapMirror operations: You follow this set of steps to set up this arrangement.

1. On the SnapVault secondary/SnapMirror source filer, use the `snapmirror store` command to copy the base snapshot of the volume to tape and use the `snapmirror use` command to continue the copy if more than one backup tape is necessary.

   For more information, see "Copying from a SnapVault volume to local tape," on page 163.

2. Use the `snapmirror update` command to manually mirror incremental updates to tape.

3. In event of data loss on the SnapVault secondary storage system, convert the volume that holds the SnapVault data from its role as a SnapMirror source volume to a destination volume, mirror the tape contents back, and then convert the restored destination volume back to a regular volume again.

   For more information, see "Restoring to SnapVault from a local tape," on page 164.

Copying from a SnapVault volume to local tape: To back up a SnapVault volume to tape, complete the following steps.

Step	Action
1	At the SnapVault secondary storage system, load the tape into a local attached tape device.
2	At the SnapVault secondary storage system, start the data transfer to tape by entering the following command: **`snapmirror store sv_volume dest_tapedevices`** *sv_volume* is the volume you are copying. *dest_tapedevices* is the tape device or a comma-separated list of tape devices to which you are copying the volume. **Result:** SnapMirror starts transferring data to tape. This command creates a baseline transfer. If you have to use more than one tape, SnapMirror prompts you for another tape.

Step	Action
3	If SnapMirror prompts you for another tape, add another tape to the drive, and continue transfer of data to tape by entering the following command: **`snapmirror use dest_tapedevices tape_drive`** *dest_tapedevices* is the tape device or a comma-separated list of tape devices to which you are copying the volume. *tape_drive* is the drive holding the new tape you supplied. **Note** The `snapmirror use` command does not support copying qtrees from tape. If you specify a qtree path as the source or destination, SnapMirror returns an error message.
4	Repeat Step 3 until SnapMirror finishes copying the volume to tape.
5	If the volume on the SnapVault secondary system that you backed up ever needs to be restored, follow the procedure described in "Restoring to SnapVault from a local tape," on page 164.

Restoring to SnapVault from a local tape: In event of data loss on a SnapVault secondary storage volume that is also configured as a SnapMirror source, data that has been SnapMirrored to tape can be restored to the SnapVault secondary storage volume with the `snapmirror retrieve` command.

Note

The `snapmirror retrieve` command restores only full volumes. If you specify a qtree path as the source or destination, SnapMirror returns an error message.

To restore a SnapMirrored volume on a SnapVault secondary storage system from tape, complete the following steps.

Step	Action
1	On the SnapVault secondary storage system, use the `vol restrict` command to put the volume that you want to restore from tape into restricted state. See the *System Administrator's Guide,* Chapter 2, for information about how to restrict a volume.

Step	Action
2	Load the tape (made with `snapmirror store`) into the local tape device.
3	Start the initialization by entering the following command on the SnapVault secondary storage system: **`snapmirror retrieve [-h]`** ***`rest_volume tape_drive`*** `The -h` option displays the headers of the source tapes but does not transfer data. *rest_volume* is the volume that you are restoring. *tape_drive* is a tape device or a comma-separated list of devices from which you are restoring the volume. **Result:** SnapMirror starts transferring data from the tape. If data is stored on more than one tape, SnapMirror prompts you for the next tape.
4	If Data ONTAP prompts you for another tape, add the next tape to the drive and restart the initialization by entering the following command: **`snapmirror use`** ***`rest_volume tape_list`*** *rest_volume* is the volume to which you are restoring. *tape_list* is the tape device from which you are restoring the volume.
5	Repeat Step 4 until SnapMirror finishes initializing the volume from the tape.

Step	Action
6	If incremental updates to the baseline snapshot data that you just restored exist on tape, insert that tape into the SnapVault secondary storage system's local tape drive and enter the following command: **`snapmirror update [options] [sv_secondary:]sv_volume`** *options* can be one or more of the following: ◆ `-k` *n* sets the maximum transfer speed to *n* kilobytes per second. This option has the same effect as the `kbs` argument in the /etc/snapmirror.conf file. ◆ `-S` [*source_filer:*]*source_volume* specifies the source filer and volume for the migration. *source_volume* is the volume you want to copy. *sv_secondary* is the name of the SnapVault secondary system that you want to restore to. *sv_volume* is the volume that you want to restore. For more information about the `snapmirror update` command, see "Updating a destination manually," on page 110.
7	After all tape-to-SnapVault retrieve and update operations are complete, use the `snapmirror break` command to turn the restored filer volume back to completely read-writable status. **`snapmirror break /vol/volume_name/`**

How SnapMirror works with the dump command

How to back up data in the destination volume

You can use the `dump` command to back up data from a SnapMirror destination volume. The `dump` command picks the most recent snapshot and copies that to tape.

You can back up any snapshot displayed by the `snap list` command on the destination. You can also create a snapshot on the source volume, copy the snapshot to the destination, and use the `dump` command to back up this snapshot from the destination to tape.

Example: `dump 0f rst0a /vol/vol1/.snapshot/weekly.0`

Effect of the dump command on the destination update schedule

Running the `dump` command on a SnapMirror destination affects SnapMirror operations on that destination in the following ways:

- Scheduled incremental SnapMirror updates of a destination *volume* can occur concurrently with a `dump` operation to tape; however, if a scheduled SnapMirror update to the destination volume involves the deletion of a snapshot that the `dump` operation is currently writing to tape, the SnapMirror update will be delayed until the `dump` operation is complete.

  Note

  SnapMirror updates of a destination *qtree* are not affected by `dump` operations under any circumstances.

- SnapMirror `break`, `resync`, and `migrate` operations cannot be carried out concurrently with the `dump` operation.

Fixing changes to SnapMirror elements

Accidental deletion of SnapMirror snapshots

SnapMirror snapshots stored on either the SnapMirror source or destination location must not be deleted.

If the base snapshot (most recent common snapshot) is accidentally deleted from either the source or destination location, attempt recovery as follows:

You might be able to recover without reinitializing the destination by breaking the SnapMirror relationship and then resynchronizing the source and the destination.

As long as there is at least one snapshot common to both the source and the destination, resynchronization will succeed. See the section on the `snapmirror break` command in "Converting a destination to a writable volume or qtree," on page 139 and the section on the `snapmirror resync` command in "Resynchronizing a SnapMirror relationship," on page 145.

If there is no snapshot common to both the source and the destination, you need to use the `snapmirror initialize` command over the network. Or, if the source and destination are volumes, use the `snapmirror store` command to store the source volume on tape and then use the `snapmirror retrieve` command to restore the volume from the tape to the destination.

If you change a destination volume name

If you change the name of a SnapMirror destination volume, you need to manually correct the SnapMirror relationships affected by the change. SnapMirror will be unable to replicate source volume data to a newly-named destination volume whose configuration information is incomplete.

In the case below, the destination, volJobak, was renamed to volStatbak. After the renaming, the `snapmirror status` command does not show the source. Instead the entry is shown with a '-' appears in the source column.

```
filerB> vol rename volJobak volStatbak
volJbak renamed to volStatbak
you may need to update /etc/exports
filerB> snapmirror status volJobak
Snapmirror is on.
filerB>snapmirror status volStatbak
Snapmirror is on.
Source        Destination         State          Lag        Status
```

```
-              filerB:volStatbak   Snapmirrored   -00:03:22  Idle
```

If you change the volume name of a SnapMirror source or destination, you need to make the following changes:

- Update the snapmirror.conf file, if there is an old entry.
- Use the `snapmirror release` command to update the old destination name, so SnapMirror will release the softlock and the old snapshot.
- Use the `snapmirror update` command on the new volume name, so status registry is updated with the new volume name.
- Update the /etc/exports file.

Caution

If a filer is running at its limit of concurrent transfers, and you attempt to initiate more transfers through manual `snapmirror update` command, the filer can panic. When this particular panic case is triggered, the system log will contain the message “snapmirror: unexpected need to notify a waiting process in replica_unconfigure”.

Data Protection Using SnapVault 5

About this chapter

This chapter discusses how the optional SnapVault on-line backup feature of Data ONTAP provides a faster, cheaper alternative or a complement to tape for data protection.

Topics in this chapter

This chapter discusses the following topics:

- "SnapVault overview" on page 172
- "Planning SnapVault" on page 180
- "Setting up a basic SnapVault backup for filer systems" on page 183
- "Setting up SnapVault backup on Open Systems platforms" on page 190
- "Managing SnapVault backup of Open Systems platforms" on page 196
- "Enabling SnapVault" on page 201
- "Starting a SnapVault backup relationship" on page 205
- "Scheduling SnapVault snapshot updates" on page 209
- "Checking SnapVault transfers" on page 214
- "Displaying SnapVault snapshots" on page 220
- "Changing settings for SnapVault backup relationships" on page 224
- "Manually updating individual secondary storage system qtrees" on page 226
- "Creating a snapshot manually" on page 228
- "Restoring SnapVault data to the primary storage system" on page 230
- "Aborting SnapVault transfers" on page 235
- "Ending SnapVault backups for a qtree" on page 237
- "Unscheduling SnapVault snapshots" on page 238
- "Releasing SnapVault relationships" on page 240
- "Protecting your SnapVault backups through SnapMirror" on page 241
- "Turning SnapVault off" on page 244

SnapVault overview

What is SnapVault

SnapVault is a disk-based storage backup feature of Data ONTAP that enables data stored on multiple filer and Open Systems (non-filer) storage systems to be backed up to a central, secondary storage system quickly and efficiently as read-only snapshots. In event of data loss or corruption on a filer or open storage system, backed up data can be restored from the SnapVault secondary storage system with less downtime and less of the uncertainty associated with conventional tape backup and restore operations.

Terminology

This chapter uses the following terms to describe the SnapVault feature:

- Primary storage system—a system whose data is to be backed up
- Secondary storage system—a filer or NearStore™ system to which data is backed up
- Primary system qtree—a qtree on a primary storage system whose data is backed up to a secondary qtree on a secondary storage system
- Secondary system qtree—a qtree on a secondary storage system to which data from a primary qtree on a primary storage system is backed up
- Open Systems platform—a non-filer system, such as a server running Solaris, HP-UX, Windows NT, or Windows 2000, whose data can be backed up to a SnapVault secondary storage system
- Open Systems SnapVault agent—a software module that can be installed on a non-filer system that enables the system to back up its data to a SnapVault secondary storage system
- SnapVault relationship—the backup relationship between a qtree on a filer primary system or a directory on an Open Systems primary platform and its corresponding secondary system qtree
- SnapVault snapshot—the backup images that SnapVault creates at intervals on its primary and secondary storage systems. SnapVault snapshots capture the state of primary qtree data on each primary system. This data is transferred to secondary qtrees on the SnapVault secondary system, which creates and maintains versions of snapshots of the combined data for long-term storage and possible restore operations.
- SnapVault snapshot basename—a name that you assign to a set of SnapVault snapshots through the `snapvault snap sched` command. As incremental snapshots for a set are taken and stored on both the primary and secondary

storage systems, the system appends a number (0, 1, 2, 3, and so on) to the basenames to track the most recent and earlier snapshot updates.

- SnapVault baseline transfer—an initial complete backup of a filer primary storage qtree or an Open Systems platform directory to a corresponding qtree on the secondary storage system
- SnapVault incremental transfer—a followup backup to the secondary storage system that contains only the changes to the primary storage data between the current and last transfer actions

Advantages of SnapVault

The SnapVault disk-based backup and restore system enables you to do the following:

- Browse backed-up files online
- Schedule fast, frequent, efficient backup of large amounts of data
- Carry out fast, flexible, selective, and simple data restore operations
- Minimize media consumption and system overhead through incremental backup
- If tape backup is necessary, offload the tape backup task from the primary storage systems to the snapshot secondary storage system which centralizes the operation and saves resources
- Configure and maintain a single system for backing up data stored on multiple platforms: Network Appliance filer, Windows 2000 server, Windows NT server, Solaris, or HP-UX systems

What data gets backed up and restored

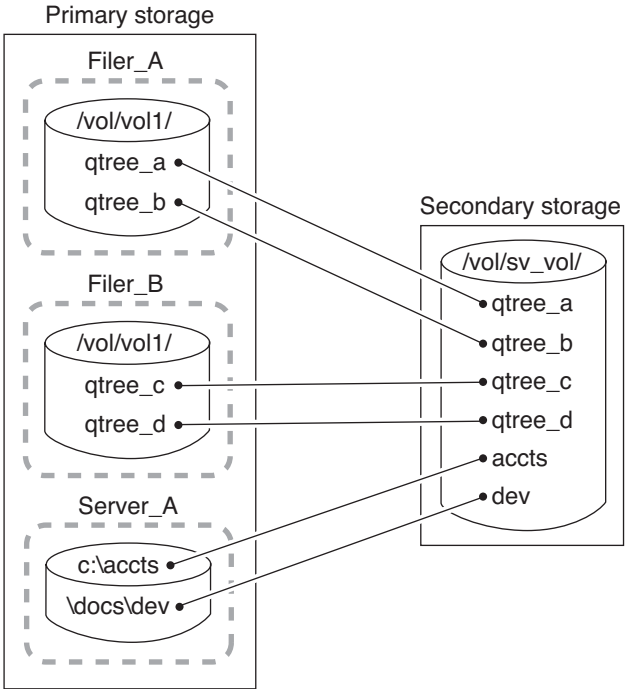
The data structures that are backed up and restored through SnapVault depend on the primary storage system.

- On NetApp filer systems, the qtree is the basic unit of SnapVault backup and restore.

  SnapVault backs up specified qtrees on the primary filer system to associated qtrees on the SnapVault secondary storage system. If necessary, data is restored from the secondary qtrees back to their associated primary qtrees.
- On Open System storage platforms, the directory is the basic unit of SnapVault backup.

  SnapVault backs up specified directories from the native system to specified qtrees in the SnapVault secondary storage system. If necessary SnapVault can restore an entire directory or a specified file to the Open System storage platform.

Primary storage qtrees from multiple filers, or primary directories from multiple Open Systems storage platforms, can all be backed up to associated qtrees on a single SnapVault secondary volume.



Basic SnapVault deployment

The basic SnapVault backup system deployment consists of the following components.

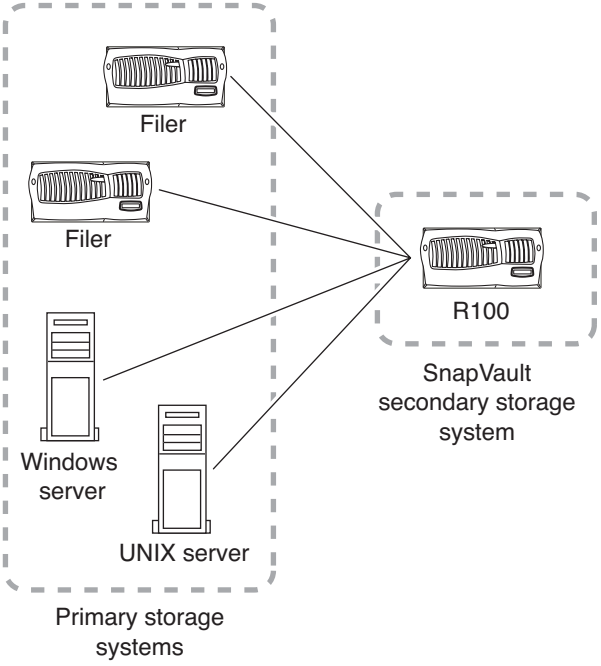
Primary storage systems: Primary storage systems are the filer and Open Systems primary storage platforms to be backed up.

- On filer primary storage systems, SnapVault backs up primary qtrees to qtree locations on the SnapVault secondary storage system.
- Supported Open Systems storage platforms include Windows NT servers, Windows 2000 servers, Solaris servers, and HP-UX servers. On Open Systems storage platforms, SnapVault can back up directories to qtree

locations on the secondary storage system. SnapVault can restore directories and single files.

Secondary storage system: The SnapVault secondary storage system is the central disk-based unit that receives and stores backup data from the filer and Open Systems storage data as snapshots. Any filer can be configured as a SnapVault secondary storage system; however, the recommended hardware platform is a Network Appliance R100 NearStore system.

The following figure shows a basic SnapVault deployment.



Primary to secondary to tape backup variation

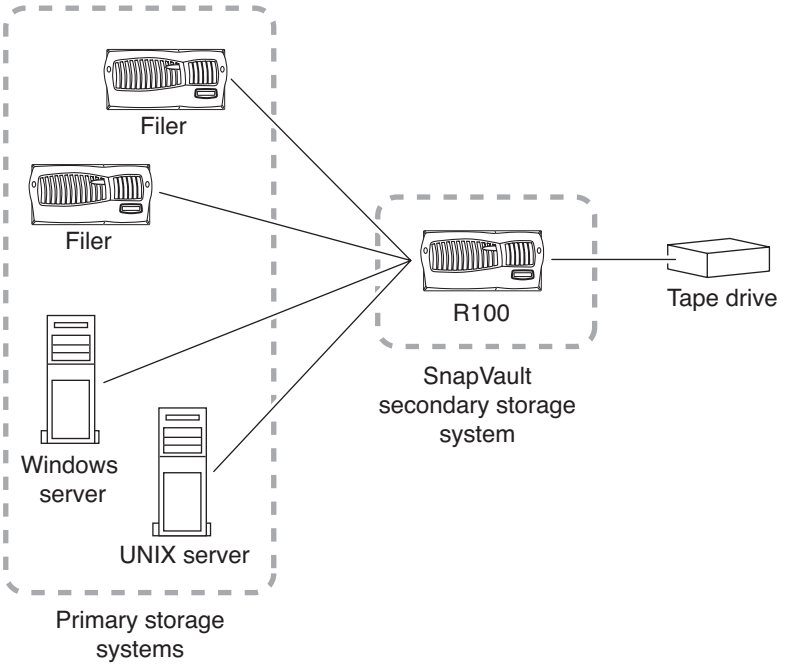
A common variation to the basic SnapVault backup deployment adds a tape backup of the SnapVault secondary storage system. This deployment might serve two purposes:

- It enables you to store an unlimited number of network backups offline while keeping your most recent backups available online in secondary storage, if necessary, for quick restoration. If you run a single tape backup off the SnapVault secondary storage system, your filer and Open Systems

storage platforms are not subject to the performance degradation, system unavailability, and complexity of direct tape backup of multiple systems.

- It can be used to restore data to a SnapVault secondary storage system in case of data loss or corruption on that system.

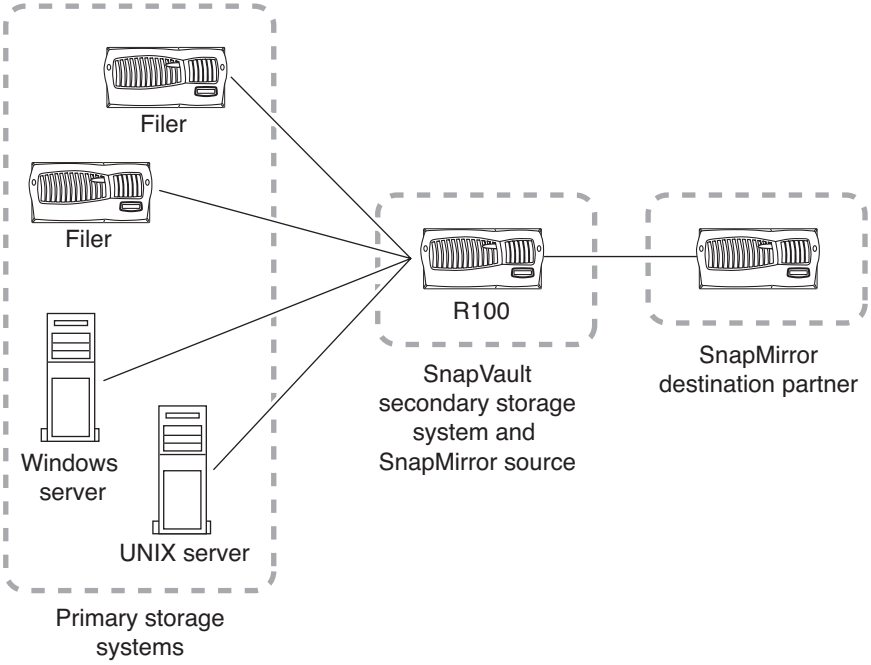
The following figure shows a basic SnapVault deployment plus tape backup.



Primary to secondary to SnapMirror variation

Another variation to the basic SnapVault deployment protects snapshots stored on SnapVault secondary storage against problems with the secondary storage system itself. The data backed up to SnapVault secondary storage is mirrored to a unit configured as a SnapMirror partner, or *destination*. If the secondary storage system fails, the data mirrored to the SnapMirror destination can be converted to a secondary storage system and used to continue the SnapVault backup operation with a minimum of disruption.

The following figure shows a SnapVault deployment with a SnapMirror backup.



How SnapVault backup of filer systems works

The process of SnapVault backup of filer qtrees is as follows:

Starting the baseline transfers:

- In response to command line input, the SnapVault secondary storage system requests initial base transfers of qtrees specified for backup from a primary storage volume to a secondary storage volume. These transfers establish SnapVault relationships between the primary and secondary qtrees.
- Each filer primary storage system, when requested by the secondary storage system, transfers initial base images of specified primary qtrees to qtree locations on the secondary storage system.

Making scheduled incremental transfers:

- Each filer primary storage system, in response to command line input, creates sets of scheduled SnapVault snapshots (which, for tracking purposes, you might name according to frequency, for example, “sv_hourly,” “sv_nightly,” and so on) of the volumes containing the qtrees to be backed up.

For each snapshot set, SnapVault saves the number of primary storage snapshots you specify and assigns each snapshot a version number (0 for most current, 1 for second most recent, and so on).

- The SnapVault secondary storage system, in response to command line input, carries out a specified set of scheduled data transfer and snapshot actions. For each of its secondary qtrees on a given volume, SnapVault retrieves, from the snapshot data of each corresponding primary qtree, the incremental changes to the primary qtrees made since the last data transfer.

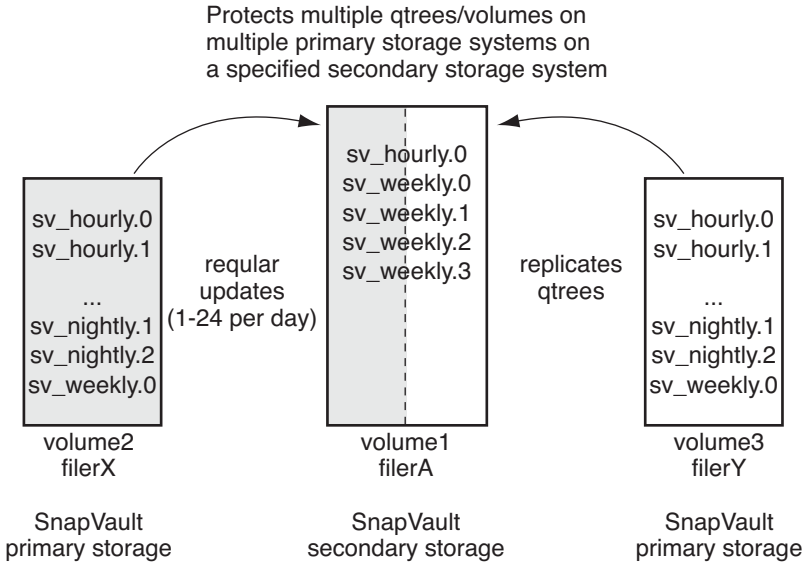
  Then SnapVault creates a volume snapshot of the changes in the secondary qtrees.

  For each transfer and snapshot set, SnapVault saves the number of secondary storage snapshots that you specify and assigns each snapshot a version number (0 for most current, 1 for second most recent, and so on).

Restoration upon request:

- If data needs to be restored to the primary storage system, SnapVault transfers the specified versions of the qtrees back to the primary storage system that requests them.

The following diagram illustrates SnapVault functionality.



How SnapVault backup of Open Systems platforms works

SnapVault backup of Open Systems platform directories works as follows:

Starting the baseline transfers:

- In response to command line input, the SnapVault secondary storage system requests from an Open Systems platform initial baseline image transfers of directories specified for backup. These transfers establish SnapVault relationships between the Open Systems platform directories and the SnapVault secondary qtrees.
- Each Open Systems platform, when prompted by the secondary storage system, transfers initial base images of specified directories to qtree locations on the secondary storage system.

Scheduling incremental transfers:

- The SnapVault secondary storage system, in response to command line input, follows a set of scheduled data transfers (to which, for tracking purposes, you can assign names like “sv_hourly,” “sv_nightly,” and so on). To each secondary qtree on a given volume, from a corresponding primary directory on the Open Systems storage platform, SnapVault transfers the files that have been added or modified since the previous data transfer.

  For each set of scheduled data transfers, SnapVault creates a set of incremental snapshots that capture the changes to the secondary qtrees after each transfer.

  For each set of snapshots, the SnapVault secondary storage system saves the number of secondary storage snapshots you specify and assigns each snapshot in the set a version number (0 for most current, 1 for second most recent, and so on).

Restore upon request:

- If directory or file data needs to be restored to the Open Systems storage platform, SnapVault retrieves the data from one of the retained snapshots and transfers the data back to the Open Systems storage platform that requests it.

Planning SnapVault

Supported primary storage platforms

Supported NetApp filer-based SnapVault primary storage platforms include:

Filer platforms: Models F8x, F7xx, F8xx, F9xx running Data ONTAP 6.2 or later and with SnapVault license purchased.

Supported primary Open Systems platforms

Supported SnapVault Open Systems primary storage platforms include:

- Windows 2000 with the Open Systems SnapVault module installed
- Windows NT with the Open Systems SnapVault module installed
- Solaris 2.6 or later with the Open Systems SnapVault module installed
- HP-UX 10.20/11 or later with the Open Systems SnapVault module installed

Supported secondary storage platforms

SnapVault secondary storage platforms that support the all the features described in this chapter include:

- NetApp NearStore R100 running Data ONTAP 6.4 and with a SnapVault secondary license purchased. This is the recommended platform.
- NetApp filer platforms: Models F8xx, and F9xx running Data ONTAP 6.4 and with a SnapVault secondary license purchased

Planning primary and secondary qtree locations

Some planning of your primary system qtrees or Open Systems directories and their corresponding secondary system qtrees is helpful. Multiple primary system qtrees from multiple volumes and multiple Open Systems directories can all be backed up to corresponding secondary system qtrees in a single volume. The maximum number of secondary system qtrees per volume is 255.

Primary system qtree or directory location example	Corresponding Secondary system qtree location example
filerA:/vol/vol1/qtreeAA	sv_secondary:/vol/sv_vol/qtreeAA
filerA:/vol/vol1/qtreeAB	sv_secondary:/vol/sv_vol/qtreeAB
filerB:/vol/vol1/qtreeBB	sv_secondary:/vol/sv_vol/qtreeBB

Primary system qtree or directory location example	Corresponding Secondary system qtree location example
winsrvrA:c:\melzdir	sv_secondary:/vol/sv_vol/melzdir
ux_srvrB:/usrs/moz_acct	sv_secondary:/vol/sv_vol/moz_acct

Planning SnapVault backup schedule and snapshot retention

On the filer-based primary systems and the SnapVault secondary storage system, the data to be backed up is captured and preserved in snapshots.

- On your filer-based primary storage systems, plan at what intervals to create SnapVault snapshots of your primary system qtrees.

  Note

  On Open Systems primary storage platforms, snapshot planning and creation does not apply.

- On your SnapVault secondary system, plan at what intervals you want to update your secondary system qtrees with data transferred from filer and Open Systems primary storage platforms, and create SnapVault snapshots to retain that information.
- Plan how to limit the combined total of snapshots retained on any one volume of the SnapVault secondary system to 251 or fewer.

  Caution

  The combined total of snapshots retained on each volume of the SnapVault secondary storage system cannot exceed 251. If the number of snapshots per volume limit is reached and the old snapshots are not deleted, SnapVault will not create new snapshots on that volume.

Before you start SnapVault configuration, use a table like the one below to plan how many snapshots you want per volume, when you want them updated, and how many of each you want to keep. For example:

- Hourly (periodically throughout the day)

  Does the data change often enough throughout the day to make it worthwhile to create a snapshot every hour or every two hours or every four hours?

- Nightly

  Do you want to create a snapshot every night or just workday nights?

- Weekly

  How many weekly snapshots is it useful to keep?

Snapshot intervals	Filer primary storage: when created	Filer primary storage: snapshots retained	Secondary storage: when created	Secondary storage: snapshots retained
weekly	sat @19	4	sat @ 21	8
nightly	mon-fri @19	10	mon-fri @20	60
hourly	@7-18	11	@8-19	120
Total	n/a	21	n/a	188

In the preceding example, the user is assumed to have 12 qtrees on the secondary storage system volume. On the secondary storage system, the user schedules

- A weekly update every Saturday at 9:00 p.m. and keeps 8 of them
- A daily update every Monday through Friday at 8:00 p.m and keeps 60 of them
- An hourly update every hour from 8:00 a.m. to 7:00 p.m. and keeps 120 of them

The result in this example is that 188 snapshots are being kept in the SnapVault secondary storage system volume.

The limit on snapshots per volume is 251, so the 188 snapshots scheduled in this example do not exceed the volume limit.

If you need to retain more than 251 snapshots: If you determine that you need to retain more than 251 SnapVault snapshots on the SnapVault secondary storage system, you can configure additional volumes on the secondary storage system. Each additional volume can support 251 additional snapshots.

Estimating the initial backup time

The backup time required for the initial transfer of data from the primary storage system to the secondary storage system depends on the inode count of the primary data to be backed up. SnapVault can carry out initial backup at an approximate rate of 7 million inodes per hour (110,000 inodes per minute).

Setting up a basic SnapVault backup for filer systems

About configuring SnapVault Backup

Setting up SnapVault backup on the filer primary systems means preparing the primary storage systems and SnapVault secondary storage systems to fulfill their backup tasks.

Note

You must have a separate SnapVault license for the primary storage and the secondary storage to use SnapVault.

- On filer primary storage systems, use filer console commands to activate the SnapVault primary license, and specify the SnapVault secondary storage host. See "Configuring a filer primary storage system for SnapVault" on page 184.
- On the SnapVault secondary storage system, use the filer console commands to license and enable SnapVault, specify the primary storage systems to back up, and start the initial snapshot backup. See "Configuring the SnapVault secondary storage system" on page 184.
- On the primary filer storage systems, schedule times for local SnapVault snapshots to occur. And on the SnapVault secondary storage system, schedule times for those primary snapshots to be backed up to secondary storage. See "Scheduling SnapVault update backups" on page 186.
- In event of data loss or corruption on a primary storage qtree, use the `snapvault restore` command to restore the affected qtree to its state at the time of its last SnapVault snapshot. See "In event of data loss or corruption" on page 188.

Note

To set up SnapVault backup on Open Systems storage platforms, see "Setting up SnapVault backup on Open Systems platforms" on page 190.

Configuring a filer primary storage system for SnapVault

On each filer primary storage system to be backed up to the SnapVault secondary storage system, log in to that filer's console and carry out the following steps.

Step	Description
1	Set up the SnapVault primary license on each primary storage system to be backed up. In the filer console, enter the following command: **`license add sv_primary_license`** For more information, see "Entering license codes" on page 201.
2	Enable SnapVault on each primary storage system to be backed up. In the filer console enter the following command:: **`options snapvault.enable on`** For more information, see "Setting the enable option" on page 202.
3	Use the `options snapvault.access` command to specify the name of the SnapVault secondary storage system to backup to. Enter the following command: **`options snapvault.access host=snapvault_secondary`** The filer must be able to resolve the host name (*snapvault_secondary)* to an IP address in the /etc/hosts file, or else the filer needs to be running DNS or NIS. Alternatively, you can use the literal IP address instead of the host name. See the na_protocolaccess(8) man page for details. For more information about the `options` command, see the na_options(1) man page. For more information, see "Setting the access option" on page 203.

Configuring the SnapVault secondary storage system

Configure the SnapVault secondary storage system. Carry out the following steps.

Step	Description
1	Set up the SnapVault secondary license. In the filer console of the SnapVault secondary system, enter the following command: **`license add sv_secondary_license`** For more information see "Entering license codes" on page 201.

Step	Description
2	Enable SnapVault. In the filer console, enter the following command: **`options snapvault.enable on`** For more information see "Setting the enable option" on page 202.
3	Use the `options snapvault.access` command to specify the names of the primary storage systems to back up and restore. Enter the following command: **`options snapvault.access host=`*`snapvault_primary1, snapvault_primary2 ...`*** For more information see "Setting the access option" on page 203.
4	For each qtree on the filer primary storage systems to be backed up, use the `snapvault start` command line to execute an initial baseline copy of the qtree from the primary to the secondary storage system. On each command line, specify the primary storage filer, volume, and qtree, and the secondary storage host, volume, and qtree. Use the `-S` prefix to indicate the source qtree path. For example, **`snapvault start -S filer_a:/vol/vol1/tree_a sv_filerb:/vol/sv_vol/tree_a`** **`snapvault start -S filer_a:/vol/vol1/tree_b sv_filerb:/vol/sv_vol/tree_b`** **`snapvault start -S filer_a:/vol/vol1/tree_c sv_filerb:/vol/sv_vol/tree_c`** **Note** At an actual console, each command is entered on a single line. For more information see "Initiating the first backup of data from filer primary storage systems" on page 205.

Scheduling SnapVault update backups

On both the filer primary storage systems and the SnapVault secondary storage system configure a snapshot schedule. To set up a snapshot schedule, complete the following steps.

Step	Description
1	**On the filer primary storage systems:** On each filer primary storage system that contains qtrees to be backed up to a SnapVault secondary storage system, use the `snapvault snap sched` command to schedule sets of SnapVault snapshots on each volume containing the backed-up qtrees. For each set of snapshots, specify volume name, snapshot basename (for example: "sv_hourly," or "sv_nightly," and so on), number of SnapVault snapshots to store locally, and the days and hours to execute the snapshots. For example: **`snapvault snap sched vol1 sv_weekly 1@sat@19`** **`snapvault snap sched vol1 sv_nightly 2@mon-fri@19`** **`snapvault snap sched vol1 sv_hourly 11@mon-fri@7-18`** **Note** When specifying the SnapVault snapshot basename, avoid using "hourly," "nightly," or "weekly." Such naming will conflict with the non-SnapVault `snap sched` snapshots. For more information, see "Scheduling snapshots on the SnapVault filer primary storage system" on page 209.

Step	Description
2	**On the SnapVault secondary storage system:** For each SnapVault volume snapshot set that you scheduled on your filer primary storage systems (see Step 1), use the `snapvault snap sched -x` command line to schedule a set of transfers to and subsequent snapshots of the SnapVault secondary storage system. For example: **`snapvault snap sched -x sv_vol sv_weekly 8@sat@20`** **`snapvault snap sched -x sv_vol sv_nightly 7@mon-fri@20`** **`snapvault snap sched -x sv_vol sv_hourly 11@mon-fri@7-19`** Snapshot basenames on the primary and secondary systems must match, but snapshot times and number of stored snapshots can differ. The `-x` parameter causes SnapVault to copy new or modified data from the primary qtrees to their associated qtrees on the secondary storage system. After all the secondary qtrees on the specified volume have been updated, SnapVault then creates a snapshot of this volume for archiving. **Note** SnapVault snapshots on the secondary system take place 5 minutes after the hour specified in the `snapvault snap sched -x` command. This delay allows time for the transfer of new data from the primary qtrees to the secondary qtrees on the specified volume. For more information see "Scheduling snapshot backups to the SnapVault secondary storage system" on page 211.

In event of data loss or corruption

In the event of data loss or corruption on a qtree in the primary storage system, the administrator can use the `snapvault restore` command to restore that qtree to its state at the time of one of its SnapVault snapshots.

Step	Description
1	In the filer console of the primary storage system whose data you want to restore, use the `snapvault restore` command to specify ◆ The path to the qtree on the secondary storage system that you want to restore from ◆ The name of the snapshot that you want to restore from (if it is not the most recent snapshot) ◆ The path on the primary storage system to the qtree you want to restore to **Example:** This example restores qtree tree_b on primary storage system melzhost from qtree tree_b on the secondary storage system, using the most recent snapshot. **`snapvault restore -S secondary_host:/vol/sv_vol/tree_b melzhost:/vol/vol1/tree_b`** **Note** At an actual console, the entire command in the example entry is entered as a single line. For more information, see "Restoring data to the primary storage system" on page 230.

Step	Description
2	After the restore operation is completed on the primary volume, go to the SnapVault secondary storage system and use the `snapvault start -r` command to restart the SnapVault backup relationship between the primary and secondary qtrees. Specify ◆ The path on the primary storage system to the qtree you want to back up ◆ The path on the secondary storage system to the qtree that you want to back up to For example: **`snapvault start -r -S melzhost/vol/vol1/tree_b secondary_host/vol/vol1/tree_b`** **Note** At an actual console, the entire command in the example entry is entered as a single line. For more information see “Restoring data to the primary storage system” on page 230.

Setting up SnapVault backup on Open Systems platforms

About configuring SnapVault backup of Open Systems storage platforms

Setting up SnapVault for Open Systems platform backup (backup of Windows NT, Windows 2000, Solaris, or HP-UX systems) involves installing special agents on those systems and preparing the SnapVault secondary storage system to fulfill its backup tasks.

- On an Open Systems platform, install the SnapVault agent appropriate to that system's platform (Windows NT, Windows 2000, Solaris, HP-UX). See "Configuring an Open Systems platform system for SnapVault" on page 190.
- On the SnapVault secondary storage system, use the filer console commands to license and enable SnapVault, specify the Open Systems platforms to back up, start the initial baseline copy, and schedule times to download updates from the Open Systems platforms. See "Configuring the Snap Vault secondary storage system for Open Systems platform backup" on page 192.
- In event of data loss or corruption on an Open Systems platform directory or file, use the `svrestore` command to restore the affected directory or file to its state at the time of its last SnapVault snapshot. See "In event of data loss or corruption on Open Systems storage" on page 195.

Configuring an Open Systems platform system for SnapVault

On each Open Systems platform (Windows NT, Windows 2000, Solaris, or HP-UNIX) to be backed up to SnapVault, log in to that system and complete the following steps.

Step	Description
1	If you have the CD-ROM labeled *Open Systems SnapVault Agent*, insert it into the target primary storage system's CD-ROM drive, and invoke the `svinstall` program from your system's command line. **`svinstall`** When the `svinstall` prompts appear, go to Step 3.

Step	Description
2	If you do not have the *Open Systems SnapVault Agent* CD-ROM, download the software module for your platform. **1.** Go to the software downloads section of the Network Appliance NOW web site, select Open Systems SnapVault, and follow the directions to download and expand the Open Systems SnapVault package appropriate to your system (Windows 2000, Windows NT, Solaris, or HP-UX). **2.** At the primary storage system command line, enter the following command. **`svinstall`**
3	When the `svinstall` program prompts for the Installation directory, press Enter to accept the default, or enter your own path and press Enter. ◆ On a Windows system the default is c:\program files\snapvault ◆ On a UNIX system, the default is /usr/snapvault
4	When the `svinstall` program prompts for the user name, password, and NDMP listening port, enter nothing and simply press Enter if you plan to manage SnapVault backup through your Data ONTAP or Open Systems platform command lines only. **Note** If you plan to manage SnapVault backup through a commercial NDMP application, specify a user and password authorized through that application.

Configuring the Snap Vault secondary storage system for Open Systems platform backup

Configure the SnapVault secondary storage system to support Open Systems platform SnapVault backup. Complete the following steps.

Step	Description
1	License the SnapVault secondary storage system. In the filer console of the SnapVault secondary system, enter the following command: **`license add sv_secondary_license`** For more information see "Entering license codes" on page 201.
2	License the SnapVault primary storage system. In the filer console of the SnapVault secondary system, enter the following command: **`license add ossv_primary_license`**
3	Enable SnapVault. In the secondary storage system filer console, enter the following command: **`options snapvault.enable on`** For more information see "Setting the enable option" on page 202.
4	Specify the names of the primary storage systems to back up to. Enter the following command: **`options snapvault.access host=snapvault_primary1, snapvault_primary2 ...`** **Example:** **`options snapvault.access host=melzhost,samzhost,budzhost`** For more information see "Setting the access option" on page 203.

Step	Description
5	For each Open Systems platform directory to be backed up to the SnapVault secondary storage system, execute an initial baseline copy from the primary to secondary storage system. ◆ Specify the fully-qualified path to the Open Systems host directory that you want to back up. Use the `-S` prefix to indicate the source path. ◆ Even though the Open Systems platform directory to be backed up has no qtree, you *still* need to specify a host and path to the qtree where you will back up this data on the SnapVault secondary storage system. Enter the following command: **snapvault start -S *prim_host*:*dirpath* *sec_host*:/vol/*sec_vol*/*sec_tree*** **Example 1 (Windows):** **snapvault start -S melzhost:c:\melzdir sv_secondary:/vol/sv_vol/tree_melz** **snapvault start -S samzhost:c:\samzdir sv_secondary:/vol/sv_vol/tree_samz** **snapvault start -S budzhost:c:\budzdir sv_secondary:/vol/sv_vol/tree_budz** **Example 2 (UNIX):** **snapvault start -S melzhost:/usr/melzdir sv_secondary:/vol/sv_vol/tree_melz** **snapvault start -S samzhost:/usr/samzdir sv_secondary:/vol/sv_vol/tree_samz** **snapvault start -S budzhost:/usr/budzdir sv_secondary:/vol/sv_vol/tree_budz** **Note** At an actual console, the command for each entry in the examples is entered as a single line.

Step	Description
6	Use the `snapvault snap sched` command to schedule update copying of new or modified data on all Open Systems platform directories that are backed up to qtrees in SnapVault secondary storage. Specify the `-x` parameter, the name of the secondary storage volume containing the secondary qtrees, a snapshot basename (for example: "sv_hourly," or "sv_nightly," and so on), the number of SnapVault snapshots to store on the secondary storage system, and the days and hours to execute. For example: **`snapvault snap sched -x vol1 sv_weekly 1@sat@19`** **`snapvault snap sched -x vol1 sv_nightly 2@mon-fri@19`** **`snapvault snap sched -x vol1 sv_hourly 11@mon-fri@7-18`** The `-x` parameter causes SnapVault to copy new or modified files from the Open Systems platform directories to their associated qtrees on the secondary storage system. After all the secondary qtrees on the specified volume have been updated, SnapVault then creates a snapshot of this volume for archiving. **Note** If files on an Open Systems platform directory are open when a scheduled SnapVault copy takes place, those files are skipped for the current transfer. Changes to those files will be backed up during the next scheduled transfer.

In event of data loss or corruption on Open Systems storage

In the event of data loss or corruption on the Open Systems platform, the administrator can use the `svrestore` command to restore a directory or file on the Open Systems platform to its state at the time of one of its SnapVault snapshots.

Step	Description
1	In the command line interface of the Open Systems platform whose data you want to restore, use the `svrestore` command to specify ◆ the secondary storage system host and the path to the secondary qtree, directory, and file that you want to restore from ◆ the name of the snapshot that you want to restore from (for example, sv_weekly.0, sv_weekly.1, or sv_weekly.2) ◆ the path on the primary storage system to the directory, or file that you want to restore to For example: **`svrestore -f sv_secondary -q /vol/sv_vol/tree_b -s sv_weekly -d c:/melzdir/evidence.doc`** **Note** In an actual command line, the entire command in this example is entered on a single line.

Managing SnapVault backup of Open Systems platforms

What SnapVault operations are possible on Open Systems platforms

SnapVault backup and restore of an Open Systems platform is carried out using many of the same commands that are used to manage backup and restore of filer primary storage systems.

In many cases, the required input for a command is different, reflecting the fact that Open Systems platform are not organized in qtrees. Instead of primary qtrees, the user specifies directories to be backed up on the Open Systems platform.

Open Systems side SnapVault commands

The following SnapVault commands can be entered at the command line of Open Systems platforms.

Command	Input for Open Systems platform
`nvpmgr`	Enables or disables SnapVault processes on the Open Systems platform. **Syntax:** At the Open Systems platform command prompt enter **`nvpmgr [shutdown]`**

Command	Input for Open Systems platform
svrestore	Restores drive, directory, or file contents to a primary storage drive or directory that was backed up to SnapVault secondary storage **Syntax:** At the Open Systems platform command prompt, enter **svrestore -f *secondary_host* -q /vol/*secondary_vol*/*secondary_tree*/*secondary_path* -d *primary_path* -s *snapshot_name*** **Example 1 (Windows):** **svrestore -f myvault -q /vol/sv_vol/melzdir\ evidence.doc -d a:\melzdir\evidence.doc -s sv_daily.0** **Example 2 (UNIX system):** **svrestore -f myvault -q /vol/sv_vol/melzdir/evidence.doc -d /usr/melzdir/evidence.doc -s sv_daily.0**

Secondary side SnapVault commands applied to Open Systems platforms

The following SnapVault commands can be entered at the command line of the SnapVault secondary storage system.

Command	Input for Open Systems platforms
license add	Specifies the license information for the SnapVault secondary storage and primary storage systems **Syntax:** At the secondary storage console, enter **license add *license_no***
options snapvault.enable	Enables SnapVault on the secondary storage system **Syntax:** At the secondary storage console, enter **options snapvault.enable {on|off}**

Command	Input for Open Systems platforms
`options snapvault.access`	Specifies the filer or Open Systems platforms that can be backed up by SnapVault **Syntax:** At the secondary storage console, enter **`options snapvault.access host=`*`prim_host`*`[,`*`prim_host`*`...]`**
`snapvault start`	Starts the initial backup of data from the Open Systems platform volumes and directories to SnapVault secondary storage qtrees **Syntax:** At the secondary storage console, enter **`snapvault start -S `*`prim_host`*`:`*`dirpath`* **`sec_host`**`:vol/`*`sec_vol`*`/`*`sec_tree`* **Example 1 (Windows):** **`snapvault start -S winservrB:c:\documents sv_hostA:vol/sv_vol/documents`** **Example 2 (UNIX):** **`snapvault start -S uxservrB:/usr/documents sv_hostA:vol/sv_vol/documents`**
`snapvault snap sched -x`	Schedules sets of incremental backups to all secondary system qtrees on a specified volume from their primary directory sources on Open Systems platforms After each incremental backup, SnapVault then creates a snapshot of updated secondary system qtrees for long-term storage. **Syntax:** At the secondary storage console, enter **`snapvault snap sched -x `*`volume_name snap_name count@weekday-weekday@hour`* **Example:** **`snapvault snap sched -x sv_vol sv_daily 5@Monday-Friday@23`**

Command	Input for Open Systems platforms
`snapvault update`	Carries out unscheduled incremental SnapVault backups from Open Systems platform drives or directories to the secondary storage qtree. **Syntax:** At the secondary storage console, enter **`snapvault update [-k kbs] [-s snapname] [sec_host:]sec_qtree_path`**
`snapvault modify`	Modifies a SnapVault relationship **Syntax:** At the secondary storage console, enter **`snapvault modify [-k kbs] [-t n] [-S [prim_host:][dir_path] [sec_host:]/vol/sec_vol/sec_qtree_path`**
`snapvault abort`	Stops an ongoing transfer of data (scheduled by the `snapvault snap sched -x` command or invoked by the `snapvault start` or `snapvault update` commands) from an Open Systems platform source to a specified secondary system qtree. **Syntax:** At the secondary console, enter **`snapvault abort [sec_filer:] sec_qtree_path`**
`snapvault stop`	Ends a SnapVault relationship between a secondary system qtree and its Open Systems platform source **Syntax:** At the secondary console, enter **`snapvault stop [sec_filer:] sec_qtree_path`**

Command	Input for Open Systems platforms
`snapvault snap unsched`	Unschedules a set of incremental backups to the secondary system qtrees on a specified volume. This set of incremental backups was originally scheduled through the `snapvault snap sched` command. This command does not end the SnapVault relationships between the secondary system qtrees and their Open Systems platform source drives or directories. **Syntax:** At the secondary storage console, enter **snapvault snap unsched [*volume_name*] [*snap_name*]** **Example:** **snapvault snap unsched sv_vol sv_daily**

Enabling SnapVault

What you need to do to enable SnapVault

You need to complete the following tasks to activate SnapVault:

- Enter the license code on both the secondary storage system and the primary storage system.
- Turn on the `options snapvault.enable` option for both the secondary storage system and the primary storage system. Setting this option enables SnapVault data transfers and snapshot creation.
- Set the `options snapvault.access` option on both the secondary storage system and the primary storage system to allow primary storage system and secondary storage system access.
  - Setting this option on the SnapVault secondary storage system determines which SnapVault primary storage systems can access the secondary storage system.
  - Setting this option on the SnapVault primary storage system determines which secondary storage system can access data from that primary storage system.

Note

The descriptions and procedures in this section pertain to SnapVault backup of NetApp filers only. For SnapVault backup of Open Systems drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 190 and "Managing SnapVault backup of Open Systems platforms" on page 196.

Entering license codes

Enter the SnapVault secondary storage system license code on the filer that is to be the secondary storage system, and enter the separate SnapVault primary storage system license code on the filer that is to be the primary storage system.

Note

You cannot license the SnapVault secondary storage system and the SnapVault primary storage system on the same filer.

To enter the SnapVault secondary storage system license code, complete the following step.

Step	Action
1	On the secondary storage system, enter the following command: **`license add xxxxxxx`** *xxxxxxx* is the license code you purchased. This setting persists across reboots.

To enter the SnapVault primary storage system license code, complete the following step.

Step	Action
1	On the primary storage system, enter the following command: **`license add xxxxxxx`** *xxxxxxx* is the license code you purchased. This setting persists across reboots.

For more information about entering license codes, see the information about licensing in Chapter 1 of the *System Administrator's Guide.*

Setting the enable option

To set the `enable` option, complete the following step.

Step	Action
1	On both the primary storage system and the secondary storage system, enter the following command: **`options snapvault.enable on`** This option persists across reboots.

Setting the access option

The `snapvault.access` option controls which filers can request data transfers. This option persists across reboots.

On the filer primary storage system: To set the primary storage systems to grant access only to the secondary storage system, complete the following step.

Step	Action
1	On the primary storage system, enter the following command: **`options snapvault.access host=`*`snapvault_secondary`* The filer must be able to resolve the host name entered as *snapvault_secondary* to an IP address in the /etc/hosts file, or else the filer needs to be running DNS or NIS. Alternatively, you can use the literal IP address instead of the host name. See the na_protocolaccess(8) man page for details. For more information about the `options` command, see the na_options(1) man page.

Example:

```
filerB> options snapvault.access host=filerA
```

On the secondary storage system: To allow the primary storage systems to restore data from the secondary storage system, complete the following step.

Step	Action
1	On the secondary storage system, enter the following command: **`options snapvault.access host=`*`snapvault_primary1, snapvault_primary2 ...`* The filer must be able to resolve the host name entered as *snapvault_primary* to an IP address in the /etc/hosts file, or else the filer needs to be running DNS or NIS. Alternatively, you can use the literal IP address instead of the host name. The syntax for specifying which filers are allowed access to the secondary storage system is described in the na_protocolaccess(8) man page. For more information about the `options` command, see the na_options(1) man page.

Example:

```
filerA> options snapvault.access host=filerB,filerC
```

Note

To grant access to any requester, enter **`options snapvault.access all`**. For security reasons, this configuration is not recommended.

Starting a SnapVault backup relationship

Why you need to specify qtrees for filer primary storage systems

After you have enabled SnapVault on both your primary and secondary storage systems and have given primary and secondary systems access to each other, you must specify the qtrees whose data you want transferred from the primary storage system to the SnapVault secondary storage device, and you must perform a complete (baseline) transfer of data from primary storage system to secondary storage.

The `snapvault start` command configures the SnapVault relationship between primary qtrees and secondary qtrees by

- Specifying the primary system qtrees to be backed up on the secondary system qtrees
- Specifying parameters for the updates from the primary qtrees to the secondary system qtrees. The parameters include transfer speed and try count (the number of times SnapVault will attempt to start a transfer before stopping the operation).
- Initiating the baseline transfer of the primary system qtree data to the secondary system qtree

Note

The descriptions and procedures in this section pertain to SnapVault backup of NetApp filers only. For descriptions and procedures pertaining to SnapVault backup of Open Systems drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 190 and "Managing SnapVault backup of Open Systems platforms" on page 196.

Initiating the first backup of data from filer primary storage systems

To specify the qtrees that you want to be backed up and to run a first-time (complete) SnapVault transfer, complete the following step.

Step	Action
1	At the filer console for the secondary storage system, enter the following command: **snapvault start [-k *kbs*] [-t *n*]** **-S *prim_filer*:*prim_qtree_path*** **[*sec_host*:]/vol/*sec_volume*/*sec_qtree_path*** `-k` *kbs* specifies a value for the throttle (transfer speed) in kilobytes per second for the primary storage system. A value of *unlimited* is the default and lets the transfer run as fast as it can. Valid values are whole positive numbers. `-t` *n* specifies the number of times SnapVault will attempt to start a transfer befor stopping the operation. The default is 2. Valid values are whole positive numbers. **Note** The `-t` parameter does not enable retries of transfers that are aborted or interrupted. `-S` [*prim_filer*:]*prim_qtree_path* specifies the qtree on the primary storage system to be backed up. This option must be set the first time the command is run for each primary storage system qtree you want to copy. *sec_host* is the name of the destination (secondary storage system) filer to which the data from the primary system qtree is transferred. If no secondary storage system filer is specified, the local host's name is used. *sec_volume* is the name of the volume on the secondary storage system to which the data from the primary storage system is transferred. *sec_qtree_path* is the path to and the name of the qtree on the secondary storage system. **Note** The qtree specified for *sec_qtree_path* must not exist on the secondary system before you run the `snapvault start` command.

Example:

```
filerB> snapvault start -S filerA:/vol/vol2/qtree3 /vol/vol1/qtree3
```

Result: SnapVault creates a snapshot of qtree3 in vol2 of the primary storage system (filerA), copies the data in it to the secondary storage system (filerB) at vol/vol1/qtree3, and configures the qtree for future updates.

Note

The time required for this baseline transfer depends on the inode count of the primary data to be backed up. SnapVault can carry out initial backup at an approximate rate of 7 million inodes per hour (110,000 inodes per minute). However, in the first phase of a large transfer SnapVault creates inodes, and it may appear that no transfer is taking place.

Initiating backup of filer primary system non-qtree data

Non-qtree data is any data on a filer that is not contained in its qtrees. Non-qtree data can include

- Configuration and logging directories on the filer (for example, /etc or /logs) that are not normally visible to filer clients
- Directories and files on a filer volume that has no qtree configured

If you want to use SnapVault to replicate and protect non-qtree data in a filer primary storage system, complete the following step.

Step	Action
1	From the secondary storage system, enter the following command: **`snapvault start -S [`*`prim_filer`*`:]/vol/`*`volume_name`*`/- /vol/`*`volume_name`*`/`*`qtree_name`*** `-S` [*prim_filer*:]/vol/*volume_name*/- specifies the volume on the primary storage system whose non-qtree data you want to backup. The dash (-) indicates all non-qtree data in the specified volume. /vol/*volume_name*/*qtree_name* specifies the qtree in the secondary storage system where you want to store this data. **Note** The qtree that is specified for /vol/*volume_name*/*qtree_name* must not exist on the secondary system before you run the `snapvault start` command.

Note

The non-qtree part of the primary storage system volume can be replicated only to the SnapVault secondary storage system. The data can be restored to a qtree on the primary storage system, but cannot be restored as non-qtree data.

Example:

```
filerB> snapvault start -S filerA:/vol/vol1/-
/vol/vol4/non_qtree_data_in_vol7
```

Result:

SnapVault transfers the non-qtree data on primary filerA to the qtree called non_qtree_data_in_vol7 in vol4 on filerB (the secondary storage system). It also configures the qtree for future updates.

Scheduling SnapVault snapshot updates

About SnapVault snapshot update schedules

After you have completed the initial baseline backup of qtrees on the filer primary storage systems to qtrees on the SnapVault secondary storage system (see "Starting a SnapVault backup relationship" on page 205), you must use the `snapvault snap sched` command to schedule

- Regular SnapVault snapshot times of volumes on the primary storage system to capture new and changed data in the qtrees that have a SnapVault relationship configured through the `snapvault start` command
- Regular transport of new or modified data in the primary qtrees to their associated secondary qtrees on the SnapVault secondary storage system
- Regular Snapshots of the volume containing the updated secondary qtrees for SnapVault archival.

Note

The descriptions and procedures in this section pertain to SnapVault backup of NetApp filers only. For descriptions and procedures pertaining to SnapVault backup of Open Systems drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 190 and "Managing SnapVault backup of Open Systems platforms" on page 196.

Scheduling snapshots on the SnapVault filer primary storage system

To schedule a set of snapshots on the SnapVault primary storage system, you use the `snapvault snap sched` command to specify the volume to be snapshot, the snapshot basename, how many versions of the snapshots to be retained, and the days and hours to execute this set of snapshots.

To set a schedule for the primary storage system, complete the following step.

Step	Action
1	From the primary storage system, enter the following command: **`snapvault snap sched volume_name snap_name schedule_spec`** *volume_name* is the name of the volume on the primary storage system on which to create this set of snapshots. *snap_name* is the basename of a snapshot set, for example, sv_nightly. The name of this snapshot must be the same on the primary storage system and on the secondary storage system. The *snap_name* must not be "hourly," "nightly," or "weekly" to avoid conflict with `snap sched` snapshots. *schedule_spec* is made up of *count*[@*day_list*][@*hour_list*]. *count* is the number of snapshots to retain for this snapshot set. A zero (0) in this field means no new instance of this snapshot will be created. **Caution** The combined total of snapshots retained for this and other snapshot sets cannot exceed 251 snapshots per volume. If it does, SnapVault will not create new snapshots. @*day_list* is a comma-separated list that specifies the days on which a new snapshot for this set is created. Valid entries are `mon tue wed thu fri sat sun`. They are not case-sensitive. You can specify a range using a dash (-), for example, `mon-fri`. The dash (-) by itself means no snapshot will be created automatically. You can create the snapshot manually. The default value is `mon-sun`. @*hour_list* specifies the hours at which a new snapshot is created for this set. Valid entries are whole numbers from 0 to 23. You can specify a range using a dash (-), or use a comma-separated list, for example, `7, 8-17, 19, 21, 23`. The default is midnight (`0`).

Example: The following three `snapvault snap sched` command lines schedule three sets of SnapVault snapshots on volume vol1 of the primary storage system.

```
filerB> snapvault snap sched vol1 sv_weekly 1@sat@19
```

```
filerB> snapvault snap sched vol1 sv_nightly 2@mon-fri@19
filerB> snapvault snap sched vol1 sv_hourly 11@mon-fri@7-18
```

Result: SnapVault primary storage system creates a snapshot on the specified volume, names it

- *sv_weekly.0* every Saturday at 7:00 p.m., and keeps one copy.
- *sv_nightly.0* every Monday through Friday at 7:00 p.m., and keeps two copies.
- *sv_hourly.0* every Monday through Friday, every hour from 7:00 a.m. to 6:00 p.m., and keeps eleven copies.

Scheduling snapshot backups to the SnapVault secondary storage system

To schedule the backup of SnapVault snapshots from the filer primary storage systems to the secondary storage system, use the `snapvault snap sched -x` command.

Complete the following step.

Step	Action
1	From the secondary storage system, enter the following command on a single line: **`snapvault snap sched -x sec_vol snap_name schedule_spec`**

In the command line just described:

- The `-x` portion of the command is required on the secondary storage system. This parameter specifies that the SnapVault secondary qtrees on the specified volume are updated from their associated primary system qtrees just before the new snapshot of the specified volume is created.
- *sec_vol* is the name of the volume on the secondary storage system for which this snapshot is scheduled.
- *snap_name* is the basename of the set of snapshots to create, for example, sv_nightly. The basename of this snapshot set must match the basename of the corresponding snapshot set configured on the primary storage system volume. Each new snapshot created for this set is numbered 0, the number of each previous snapshot in this set is increased by 1, and the oldest snapshot in this set is deleted. The *snap_name* must not be “hourly,” “nightly,” or “weekly” to avoid conflict with regular Data ONTAP `snap sched` snapshots.
- *schedule_spec* is made up of *count*[@*day_list*][@*hour_list*].
  - *count* is the number of snapshots to retain for this set. A zero (0) in this field means no new secondary storage system snapshot will be created

for this set, although the qtrees on the secondary storage system will be updated by the transfers from the primary storage systems.

Caution

The combined total of snapshots retained for this and other snapshot sets cannot exceed 251 snapshots per volume. If it does, SnapVault will not create new snapshots.

- *@day_list* is a comma-separated list that specifies the days on which a new snapshot is created for this set. Valid entries are `mon tue wed thu fri sat sun`. They are not case-sensitive. You can specify a range using a dash (-), for example, `mon-sun`. The dash (-) by itself means no snapshot will be created automatically. You can create the snapshot manually.
- *@hour_list* specifies the hours at which a new snapshot is created for this set. Valid entries are whole numbers from 0 to 23. You can specify a range using a dash (-), or use a comma-separated list, for example, `6, 8-17, 19, 21, 23`. The default is midnight (`0`).

Note

You can stop and unschedule a set of snapshots on the primary storage system or the secondary storage system at any time with the `snapvault snap unsched` command. (See "Unscheduling SnapVault snapshots" on page 238.)

Example: The following three `snapvault snap sched` command lines schedule three sets of SnapVault transfer updates and snapshots on volume vol1 of the secondary storage system.

```
filerA> snapvault snap sched -x vol1 sv_weekly 5@sat@21
filerA> snapvault snap sched -x vol1 sv_nightly 5@mon-fri@20
filerA> snapvault snap sched -x vol1 sv_hourly 4@mon-fri@8-19
```

Result: SnapVault transfers qtree data from the primary storage system's snapshot as follows:

- SnapVault transfers *sv_weekly.0* to the secondary storage system every Saturday at 9:00 p.m., makes a new snapshot with the same name of all the transferred data, and keeps five copies.
- SnapVault transfers *sv_nightly.0* to the secondary storage system every Monday through Friday at 8:00 p.m., makes a new snapshot with the same name of all the transferred data, and keeps five copies.
- SnapVault transfers *sv_hourly.0* to the secondary storage system every hour from 8:00 a.m. to 7:00 p.m., Monday through Friday, makes a new snapshot with the same name of all the transferred data, and keeps four copies.

Note

Snapshots scheduled on the secondary storage system with the `snapvault snap sched -x` command are created five minutes after the hour you specify to give SnapVault on the filer primary storage systems enough time to create the snapshots they need before the secondary storage system updates from them.

How to display the currently configured snapshot schedule

Enter the `snapvault snap sched` command without a schedule specification to show the current schedule values. If no *snap_name* is given, the command shows the basenames of all snapshot sets scheduled for the specified volume. If no *volume_name* is given, the command shows the basenames of all snapshots sets for all SnapVault volumes on the filer.

Example: `filerA> snapvault snap sched`

```
xfer   vol1 sv_weekly 5@sat@21
xfer   vol1 sv_nightly 5@mon-fri@20
xfer   vol1 sv_hourly 4@mon-fri@8-19
xfer   vol2 sv_nightly 5@mon-fri@20
xfer   vol2 sv_hourly 4@mon-fri@8-19
```

Checking SnapVault transfers

Why to check data replication status

You check transfer status, using the `snapvault status` command, to make sure SnapVault transfers are taking place as they should.

Note

The descriptions and procedures in this section pertain to SnapVault backup of NetApp filers only. For descriptions and procedures pertaining to SnapVault backup of Open Systems drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 190 and "Managing SnapVault backup of Open Systems platforms" on page 196.

Checking the status

To check the status of a data transfer and see how recently a qtree has been updated, complete the following step.

Step	Action
1	Enter the following command: **`snapvault status [-l] [[[filer_name:]qtree_path] ...] [-s] [-c]`** Options can be one or more of the following: ◆ `-l` displays the long format of the output, which contains more detailed information. ◆ `-s` displays the SnapVault snapshot basename, status, and schedule for each volume. ◆ `-c` displays the configuration parameters of all SnapVault qtrees on the filer. This option can be run only from the secondary storage system. *filer_name* is the name of the filer for which you want to see the status of SnapVault operations. *qtree_path* is the path of the qtree or qtrees for which you want to see the status of SnapVault operations. You can specify more than one qtree path.

Result: The filer displays a message showing whether a transfer is in progress, how much data has been transferred, the state of the destination, and how long ago the last successful transfer took place.

- If no options or arguments are given, the output includes the fields of information in horizontal format shown in Example 1 below.

  If one or more [*filer_name*:]*qtree_path* arguments are specified, then status is displayed only for the specified qtrees.
- If the `-l` option is given, the output includes the more detailed information, in vertical format, shown in Example 2.

  See the table "What the status fields mean" on page 216, after the examples, for more information about the command output shown in Example 1 and Example 2.
- If the `-s` option is given, the output displays snapshot creation status, as shown in Example 3.
- If the `-c` option is given, current primary storage system configuration parameter settings are displayed, as shown in Example 4.

Example 1: If you enter `snapvault status` with no option you see SnapVault qtree relationships involved in the most recent transfer from a primary qtree to a secondary qtree.

```
filerA> snapvault status
Snapvault client is ON.
```

```
Source                   Destination                    State         Lag       Status
filerB:/vol/vol2/qtree3  filerA:/vol/sv_vol/qtree3      Snapvaulted   00:48:24  Idle
```

Example 2: The `snapvault status -l` command and option displays more detailed information on the most recent SnapVault transfer and snapshot activity.

```
filerA> snapvault status -l
SnapVault client is ON.

Source:                     filerA:/vol/vol2/qtree3
Destination                 filerB:/vol/sv_vol/qtree3
Status                      Idle
Progress:                   -
State:                      SnapVaulted
Lag:                        2:09:58
SnapVault Timestamp:        Thu Jan 31 12:30:01 PST 2002
Base Snapshot:              filerB(0016778780)_sv_vol.59
Current Transfer Type
```

```
Current Transfer Error:
Contents:                   Replica
Last Transfer Type:         Scheduled
Last Transfer Size:         1680 KB
Last Transfer Duration:     00:02:08-
```

Example 3: The `snapvault status -s` command and option lists all the snapshots scheduled on the primary or secondary storage system. Information includes volume, snapshot basename, current status, and snapshot schedule.

filerA> snapvault status -s

```
Snapvault server is ON.

Volume          Snapshot      Status              Schedule
------          --------      ------              --------
sv_vol          sv_hourly     Idle                1@0-23
vol2            sv_weekly     Idle                8@fri
```

Example 4: The snapvault status `-c` command and option lists all the secondary system qtrees, their corresponding primary system qtrees, maximum speed of scheduled transfers, and maximum number of times SnapVault attempts to start a scheduled transfer before skipping that transfer.

```
filerA> snapvault status -c
/vol/sv_vol/db_qtree1 source=filerB:/vol/db1/s1 kbs=unlimited
tries=20
/vol/sv_vol/db_qtree2 source=filerC:/vol/db2/s2 kbs=unlimited
tries=20
/vol/sv_vol/qtree1 source=filerC:/vol/users/qtree1 kbs=10000
tries=10
/vol/sv_vol/qtree2 source=filerC:/vol/users/qtree2 kbs=10000
tries=10
/vol/sv_vol/qtree3 source=filerD:/vol/users/qtree1 kbs=10000
tries=10
/vol/sv_vol/qtree4 source=filerD:/vol/db/db_1 kbs=7000 tries=10

filerA> snapvault status -c /vol/sv_vol/qtree3
/vol/sv_vol/qtree3 source=filerA:/vol/vol2/qtree3 kbs=10000
tries=10
```

What the status fields mean

Information fields that SnapVault can display for the `snapvault status` and `snapvault status -l` commands are as follows.

Field	Possible values displayed include:
Source	*filer:qtree_path*—The source is the primary storage system filer and qtree path listed. A dash (-) in this field means that the SnapVault secondary storage destination qtree is not yet initialized through the `snapvault start -x` command.
Destination	*filer:qtree_path*—The destination is the secondary storage system filer and qtree path listed.
State	Uninitialized—The destination SnapVault secondary storage qtree is not yet initialized through the `snapvault start` command. Snapvaulted—The qtree is a SnapVault secondary destination. Unknown—It is not known what state the secondary system qtree is in; the volume that contains the secondary system qtree could be offline or restricted. Source—This state is reported when the `snapvault status` command is run on the primary storage system. It also appears if `snapvault status` is run on secondary storage systems after the `snapvault restore` command was run on an associated primary storage system.
Lag	*hh:mm:ss* indicates the time difference between the data currently on the primary system and the latest data stored on the secondary system; that is, the difference between the current time and the timestamp of the snapshot last successfully transferred to the secondary storage system. A dash (-) in this field means that the secondary storage system is not initialized.

Field	Possible values displayed include:
Status	idle—No data is being transferred. Transferring—Transfer has been initiated, but has not yet started, or is just finishing. Pending—The secondary storage system cannot be updated because of a resource issue; the transfer is retried automatically. Aborting—A transfer is being aborted and cleaned up. Resyncing—The specified qtree is resynchronizing.
Progress	Shows the number of KB transferred by the current transfer or the restart check point if the status is Idle or Pending.
SnapVault Timestamp	The timestamp of the last snapshot successfully transferred from the primary storage system to the secondary storage system. **Note** A resynchronization (`snapvault start -r`) may change the base snapshot to a snapshot with a timestamp older than the original base.
Base Snapshot	The name of the base snapshot for the corresponding qtree on the secondary storage system. For qtrees in a SnapVault relationship, the secondary storage side lists the name of the exported snapshot for that qtree on the storage side. A resynchronization (`snapvault start -r`) may change the name of the base snapshot.
Current Transfer Type	Indicates the type of the current transfer: scheduled, retry, resync, update, initialize, store, or retrieve. This field applies only to the destination side.
Current Transfer Error	Displays an error message if the latest transfer attempt failed.

Field	Possible values displayed include:
Contents	Indicates whether the contents of the destination volume or qtree in the active file system are up-to-date replicas or are in transition. The field applies only to the destination side. Since a destination is read-only, its contents are always a replica.
Last Transfer Type	Indicates the type of the previous transfer: scheduled, retry, resync, update, initialize, store, or retrieve. This field applies only to the secondary storage side.
Last Transfer Size	Shows the number of KB transferred in the last successful transfer.
Last Transfer Duration	Shows the elapsed time for the last successful transfer. If the transfer failed and restarted, the time includes time waiting to restart the transfer. If a transfer aborted and was retried from the beginning, it includes only the time required for the final successful attempt.
Last Transfer From	This field applies only to the secondary storage side and shows the name of the primary filer and volume or qtree (where the content is transferred from).

Displaying SnapVault snapshots

Why to display SnapVault snapshots and qtrees

You can use the `snap list` command to display a list of snapshots to confirm what versions of your primary qtree data have been backed up, or to locate by date or time a particular version of a qtree to retrieve. Using the `snap list -q` command, you can see the following:

- A list of all snapshots on the secondary storage system (not just SnapVault snapshots)
- The qtrees in the snapshots
- The primary storage system sources of those qtrees
- The timestamp of the primary storage system snapshot that was the source for the data in the secondary storage system snapshot

Using the `snap list -o` command, you can also list the snapshot timestamps, primary qtree origin (if applicable) and snapshot names stored for an individual qtree.

Note

The descriptions and procedures in this section pertain to SnapVault backup of NetApp filers only. For descriptions and procedures pertaining to SnapVault backup of Open Systems drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 190 and "Managing SnapVault backup of Open Systems platforms" on page 196.

How to display SnapVault snapshots on volumes

To see a list of the snapshots and qtrees on your filer volumes, complete the following step.

Step	Action
1	On the filer for which you want to see the information, enter the following command: **`snap list -q [`*`volume_name`*`]`** *volume_name* is the name of the volume for which you want to list the snapshots. If no volume name is given, the snapshots on all this filer's volumes are displayed.

Sample snap list -q output

Primary storage output example: If you specify the primary volume name, the command lists the information for each snapshot in the volume. This output is from a volume used as a SnapVault primary storage system.

```
filerA> snap list -q vol2
working...

Volume vol2
    qtree                contents       timestamp     source
    -----                --------       ---------     ------
sv_hourly.0  (Jan 22 20:00)
    qtree1               Original       Jan 22 20:00  -
    qtree2               Original       Jan 22 20:00  -
    qtreeZ               Original       Jan 22 20:00  -
sv_hourly.1  (Jan 22 16:00)
    qtree1               Original       Jan 22 16:00  -
    qtree2               Original       Jan 22 16:00  -
    qtreeZ               Original       Jan 22 16:00  -
sv_hourly.2  (Jan 22 12:00)
    qtree1               Original       Jan 22 12:00  -
    qtree2               Original       Jan 22 12:00  -
    qtreeZ               Original       Jan 22 12:00  -
sv_hourly.3  (Jan 22 08:00)
    qtree1               Original       Jan 22 08:00  -
    qtree2               Original       Jan 22 08:00  -
    qtreeZ               Original       Jan 22 08:00  -
sv_nightly.0  (Jan 22 00:00)
    qtree1               Original       Jan 22 00:00  -
    qtree2               Original       Jan 22 00:00  -
    qtreeZ               Original       Jan 22 00:00  -
sv_hourly.4  (Jan 21 20:00)
    qtree1               Original       Jan 21 20:00  -
    qtree2               Original       Jan 21 20:00  -
    qtreeZ               Original       Jan 21 20:00  -
sv_hourly.5  (Jan 21 16:00)
    qtree1               Original       Jan 21 16:00  -
    qtree2               Original       Jan 21 16:00  -
    qtreeZ               Original       Jan 21 16:00  -
sv_nightly.1  (Jan 21 00:00)
    qtree1               Original       Jan 21 00:00  -
    qtree2               Original       Jan 21 00:00  -
    qtreeZ               Original       Jan 21 00:00  -
```

This output shows which qtrees were writable and therefore have original content (the timestamp in these cases is the same as for the snapshot as a whole). It also shows whether any qtrees were transitioning and are therefore neither a faithful replica nor original content. Transitioning qtrees are shown with a dash (-) instead of a timestamp.

Secondary storage output example: If you specify the volume name (in this example, sv_vol) and are running the command from a filer used as a SnapVault secondary storage system, you see a list of all the SnapVault snapshots retained on volume sv_vol and the details of the qtrees contained in those snapshots.

```
filerB> snap list -q sv_vol
Volume sv_vol
working...

   qtree       contents    date           source
   --------    ---------   --------       -------
sv_hourly.0 (Jan 31 20:00)
   qtree1      Replica     Jan 22 20:40   filerA:/vol/vol2/qtree1
   qtree2      Replica     Jan 22 20:40   filerA:/vol/vol2/qtree2
   qtreeZ      Replica     Jan 22 20:40   filerA:/vol/vol2/qtreeZ
sv_hourly.1  (Jan 22 16:00)
   qtree1      Replica     Jan 22 16:00   filerA:/vol/vol2/qtree1
   qtree2      Replica     Jan 22 16:00   filerA:/vol/vol2/qtree2
   qtreeZ      Replica     Jan 22 16:00   filerA:/vol/vol2/qtreeZ
sv_hourly.2  (Jan 22 12:00)
   qtree1      Replica     Jan 22 12:00   filerA:/vol/vol2/qtree1
   qtree2      Replica     Jan 22 12:00   filerA:/vol/vol2/qtree2
   qtreeZ      Replica     Jan 22 12:00   filerA:/vol/vol2/qtreeZ
.....
```

This output shows which qtrees are replicas of another qtree, and the timestamp of the source snapshot.

Note

Qtrees that are transitioning appear with a dash (-) instead of a timestamp. In general, do not attempt to restore snapshot versions of qtrees that are transitioning.

If you specify no arguments, the output shows the information for each snapshot in each volume.

How to list snapshots for qtrees

To see a list of snapshots associated with a qtree and, if applicable, the snapshots' primary qtree origins, complete the following step.

Step	Action
1	On the filer for which you want to see the information, enter the following command: **snap list -o [*qtree_path*]** *qtree_path* displays one qtree. If no qtree name is given, information for all the qtree names on the volume are displayed.

Sample snap list -o output

If you specify the `-o` parameter with a qtree path, the `snap list` output includes the dates, sources (if any), and name of associated SnapVault snapshots that are retained on the system, for example:

```
filerB> snap list -o /vol/sv_vol/qtree3
working...

Qtree /vol/sv_vol/qtree3
date          source                       name
------------  --------                     --------
Jan 31 18:00  filerA:/vol/vol2/qtree3      hourly.0
Jan 31 17:00  filerA:/vol/vol2/qtree3      hourly.1
Jan 31 16:00  filerA:/vol/vol2/qtree3      hourly.2
Jan 31 15:00  filerA:/vol/vol2/qtree3      hourly.3
Jan 30 14:00  filerA:/vol/vol2/qtree3      hourly.4
Jan 30 13:00  filerA:/vol/vol2/qtree3      hourly.5
Jan 30 12:00  filerA:/vol/vol2/qtree3      hourly.6
Jan 30 11:00  filerA:/vol/vol2/qtree3      hourly.7
Jan 31 10:00  filerA:/vol/vol2/qtree3      hourly.8
Jan 31 9:00   filerA:/vol/vol2/qtree3      hourly.9
Jan 31 8:00   filerA:/vol/vol2/qtree3      hourly.10
Jan 30 20:00  filerA:/vol/vol2/qtree3      nightly.0
Jan 29 20:00  filerA:/vol/vol2/qtree3      nightly.1
Jan 26 16:00  filerA:/vol/vol2/qtree3      weekly.0
```

Changing settings for SnapVault backup relationships

Why you change SnapVault settings

You can use the `snapvault modify` command to change the primary storage system (source) qtree that you specified using the `snapvault start` command.

You can also use the `snapvault modify` command to change the SnapVault settings for transfer speed and number of tries before quitting that you entered using the `snapvault start` command, if there are hardware or software changes to your filers.

Note

The descriptions and procedures in this section pertain to SnapVault backup of NetApp filers only. For descriptions and procedures pertaining to SnapVault backup of Open Systems drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 190 and "Managing SnapVault backup of Open Systems platforms" on page 196.

The `snapvault modify` command is available only from the secondary storage system.

If you need to change the SnapVault schedule, use the `snapvault snap sched` command, explained in more detail in "Starting a SnapVault backup relationship" on page 205.

How to change SnapVault settings

To change the values you entered with the `snapvault start` command, complete the following step.

Step	Action
1	From the secondary storage system, enter the following command on a single line: **snapvault modify [-k *kbs*] [-t *n*] [-S [*prim_filer*:]*prim_qtree_path*] [*sec_filer*:]*sec_qtree_path*** -k *kbs* specifies a value in kilobytes per second for the throttle (transfer speed) for the primary storage system. A value of *unlimited* lets the transfer run as fast as it can. Other valid values are whole positive numbers. -t *n* specifies the number of times to try the transfer before giving up. The default is 2. Valid values are positive integers. -S [*prim_filer*:]*prim_qtree_path* specifies the primary storage system for the qtree. [*sec_filer*:]*sec_qtree_path* specifies the secondary storage system for the update.

Example: This example shows how to modify SnapVault so that it will continue backing up the data on the primary storage system filerB:/vol2/qtree3 after the name of qtree3 on the primary storage system changes to qtreeBob.

```
filerA> snapvault status

Snapvault server is ON.
Source                    Destination                 State           Lag        Status
filerB:/vol/vol2/qtree3   filerA:/vol/sv_vol/qtree3  Snapvaulted    02:48:24   Idle
```

[The qtree3 on filerB is renamed qtreeBob.]

```
filerA> snapvault modify -S filerB:/vol/vol2/qtreeBob /vol/sv_vol/qtree3

filerA> snapvault status
Snapvault server is ON.
Source                     Destination                 State          Lag       Status
filerB:/vol/vol2/qtreeBob  filerA:/vol/sv_vol/qtreeBob  Snapvaulted  0:00:31   Idle
```

Manually updating individual secondary storage system qtrees

Why to manually update a qtree on the secondary storage system

You can use the `snapvault update` command to manually update the SnapVault qtree on the secondary storage system from a snapshot on the primary storage system. You might want to update at an unscheduled time to protect the primary storage system data because

- A disk failed on the primary storage system and you want extra protection for the data
- The nightly backup failed due to a network problem
- The primary storage system hardware is going to be reconfigured
- You want to transfer a snapshot of a quiesced database

Note

The descriptions and procedures in this section pertain to SnapVault backup of NetApp filers only. For SnapVault backup of Opens System drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 190 and "Managing SnapVault backup of Open Systems platforms" on page 196.

How to update the snapshot on the secondary storage system

To manually update a SnapVault qtree on the secondary storage system, complete the following step.

Step	Action
1	Enter the following command from the secondary storage system: **`snapvault update [`*`options`*`] [`*`sec_filer`*`:]`*`sec_qtree_path`* *options* can be one or more of the following: ◆ `-k` *kbs* overrides the configured rate and specifies a value in kilobytes per second for the throttle (transfer speed) for this snapshot transfer. The default value, *unlimited,* lets the transfer run as fast as it can. ◆ `-s` *snapname* enables you to specify a primary storage system snapshot that is more recent than the current base snapshot. [*sec_filer*:]*sec_qtree_path* is the name of the secondary storage system qtree that you want to update.

Example 1: `filerB> snapvault update /vol/vol2/qtree3`

Result: SnapVault updates the qtree on the secondary storage system (filerB) with the data from a new snapshot of the qtree it creates on the primary storage system (filerA). You do not have to specify where the primary storage system data is, because you already did so when you set up the SnapVault relationship using the `snapvault start` command.

Example 2: To update qtree3 on the secondary storage system (filerB) with a particular snapshot of qtree3 on the primary storage system (filerA), you would use the -s option to specify the snapshot on the primary storage system, as in the following example.

```
filerB> snapvault update -s my_snap filerB:/vol/vol0/qtree3
```

Result: SnapVault updates the qtree on the secondary storage system (filerB) with the data from a snapshot of qtree3 (my_snap) that you created earlier on the primary storage system (filerA).

Note

The `snapvault update` command does not create a new snapshot on the secondary storage system. Use the `snapvault snap create` command if you want to create a new snapshot on the secondary storage system. See "Creating a snapshot manually" on page 228.

Creating a snapshot manually

Why to create a snapshot manually

You might create a manual (unscheduled) snapshot for the following reasons:

- You anticipate planned downtime or you need to recover from downtime (during which a snapshot was not taken on time).
- You have just carried out a manual update of a secondary qtree, as described in "Manually updating individual secondary storage system qtrees" on page 226, and you want to immediately incorporate that update into the retained snapshots on the secondary storage system.

Note

The descriptions and procedures in this section pertain to SnapVault backup of NetApp filers only. For descriptions and procedures pertaining to SnapVault backup of Open Systems drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 190 and "Managing SnapVault backup of Open Systems platforms" on page 196.

How to create a snapshot manually

You run this command on the SnapVault primary storage system or SnapVault secondary storage system containing the volume on which you want to create the snapshot. To create a manual snapshot of a volume, complete the following step.

Step	Action
1	From the primary storage system or secondary storage system, enter the following command: **`snapvault snap create volume_name snap_name`** *volume_name* is the name of the volume where the snapshot to be created will reside. *snap_name* is the basename of the snapshot to create. If there is already a snapshot being created in the volume at the time this command is invoked, this command will be carried out after the other snapshot is completed.

Example: `filerB> snapvault snap create vol1 sv_nightly`

Result: SnapVault creates a new snapshot and, based on the specified snapshot basename, numbers it just as if that snapshot had been created by the SnapVault schedule process. SnapVault names the new snapshot sv_nightly.0, renames the older snapshots, and deletes the oldest sv_nightly snapshot.

Note

Unlike the `snapvault snap sched -x` command, the `snapvault snap create` command does not update the data in the secondary storage system qtree from the data in the primary storage system qtree prior to creating the new snapshot. If you want to update your secondary qtrees prior to running the `snapvault snap create` command, use the `snapvault update` command as described in "Manually updating individual secondary storage system qtrees" on page 226.

Restoring SnapVault data to the primary storage system

About SnapVault data restoration

In the event of data loss on a primary storage system, restoring data from the SnapVault secondary storage system involves these command line operations.

- You use the `snapvault restore` command to restore a backed-up qtree from its last update saved to the secondary storage system.
- After successfully restoring data, you use the `snapvault restart` command to restart the SnapVault backup relationship between the primary and secondary qtrees (because you want to continue SnapVault protection of the data). If you do not want the continue the backup relationship, you use the `snapvault release` command to cancel any further backups of the restored qtree and to release the resources on the secondary storage system that were used in the SnapVault relationship.

Note

The descriptions and procedures in this section pertain to SnapVault backup of NetApp filers only. For SnapVault backup of Open Systems drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 190 and "Managing SnapVault backup of Open Systems platforms" on page 196.

Restoring data to the primary storage system

To restore data from the SnapVault secondary storage system to the primary storage system, complete the following step.

Step	Action
1	If you intend to restore a primary qtree to the exact qtree location on the primary storage system from which you backed it up, delete the existing qtree from the primary storage system.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>2</td><td colspan="2">Enter the following command on the primary storage system on a single line:
<tt>snapvault restore [options] -S [sec_filer:]sec_qtree_path [prim_filer:]prim_qtree_path</tt>
options can be one or more of the following:
◆ <tt>-s</tt> snapname specifies the name of the secondary storage system snapshot that you want to restore to the primary storage system, if it is not the most recent snapshot.
◆ <tt>-k</tt> kbs specifies a value for the throttle (transfer speed) for this snapshot in kilobytes per second. A value of unlimited is the default and lets the transfer run as fast as it can.
<tt>-S</tt> [sec_filer:]sec_qtree_path specifies the secondary storage system and qtree path from which you want to restore the data.
prim_filer is the name of the primary storage system filer that you want to restore to. If specified, this filer name must match the name of the host filer.
prim_qtree_path is the name of the primary storage system qtree that you want to restore to.</td></tr>
<tr><td rowspan="3">3</td><td>If you want to...</td><td>Then...</td></tr>
<tr><td>Resume SnapVault backups of the newly restored qtree on the primary storage system</td><td>Go to Step 4.</td></tr>
<tr><td>Discontinue the backup relationship between the newly restored qtree on the primary storage system and its secondary qtree partner</td><td>Go to Step 6.</td></tr>
</table>

Step	Action
4	If you want to resume the SnapVault relationship between the restored qtree and its backup qtree on the secondary storage system, enter the following command on a single line: **`snapvault start -r [`*`options`*`] -S [`*`prim_filer`*`:]`*`prim_qtree_path`* **`[`*`sec_filer`*`:]/vol/`*`sec_volume`*`/`*`sec_qtree_path`* `-r` is required to restart backups from a restored primary storage system. *options* can be one or more of the following: ◆ `-k` *kbs* sets a value for the throttle (transfer speed) in kilobytes per second for the primary storage system. ◆ `-t` *n* sets the number of times to try to start the transfer before giving up. (This parameter does not enable SnapVault to restart transfers that have started but failed to complete.) `-S` [*prim_filer*:]*prim_qtree_path* specifies the primary storage system and path for the restored qtree. *sec_filer* is the name of the secondary storage system filer to which the snapshots from the primary storage system are transferred. If specified, this filer name must match the name of the host filer. If no secondary storage system filer is specified, the local host's name is used for the filer name. *sec_volume* is the name of the volume on the secondary storage system to which the data from the primary storage system is transferred. *sec_qtree_path* is the path and name of the qtree on the secondary storage system. It must be the same name and path as used for the *sec_qtree_path* variable in Step 1.
5	You have finished. See Example 1.

Step	Action
6	If you want to discontinue the SnapVault relationship between the restored qtree and its backup qtree on the secondary storage system, enter the following command on the secondary storage system: **snapvault release *sec_qtree_path* [*prim_filer*:]*prim_qtree_path*** *sec_qtree_path* is the name of the secondary storage system qtree that you want to release from a SnapVault relationship. [*prim_filer*:]*prim_qtree_path* is the name of the primary storage system qtree that you want to release.
7	You have finished. See Example 2.

Example 1: This example shows the primary storage system (filerA) requesting data to be restored from the secondary storage system (filerB) and then the secondary storage system restarting the SnapVault backup relationship.

```
filerA> snapvault restore -S filerB:/vol/sv_vol/qtree3
vol/vol1/qtree3

filerB> snapvault start -r -S filerA:/vol/vol1/qtree3
filerB:/vol/sv_vol/qtree3
```

Example 2: This example shows the primary storage system (filerA) requesting data to be restored from the secondary storage system (filerB) and then the secondary storage system canceling the SnapVault backup relationship on both filers to release the resources used.

```
filerA> snapvault restore -S filerB:/vol/sv_vol/qtree3
/vol/vol1/qtree3

filerB> snapvault release /vol/sv_vol/qtree3
filerA:/vol/vol1/qtree3
```

Deleting the residual snapshot

When you use the `snapvault restore` command to restore a primary qtree, SnapVault places a residual SnapVault snapshot on the volume of the restored primary qtree. This snapshot is not automatically deleted. If you have configured this volume to retain the maximum 251 snapshots allowed by Data ONTAP, you must manually delete this residual snapshot.

To find and delete this residual snapshot, complete the following steps.

Step	Action
1	To display all snapshots (including the residual snapshot) on the volume of the restored qtree, enter the following command: **`snap list primaryvolume`** The residual snapshot will be distinguished by the following syntax: *`primaryhost`* `(`*`nvram_id`*`)_`*`primaryvolume_restoredqtree`*`-dst.2` **Example:** `prim_filer (1880911275)_vol1_mytree-dst.2`
2	To delete the residual snapshot, enter the following command: `snap delete` *`primaryvolume extrasnapshotname`* **Example:** **`snap delete vol1 prim_filer (1880911275)_vol1_mytree-dst.2`**

Aborting SnapVault transfers

Why you abort transfers

You can use the `snapvault abort` command to halt an ongoing SnapVault transfer if a later transfer would be more useful or if an immediate shutdown or reboot is necessary.

Note

The descriptions and procedures in this section pertain to SnapVault backup of NetApp filers only. For descriptions and procedures pertaining to SnapVault backup of Open Systems drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 190 and "Managing SnapVault backup of Open Systems platforms" on page 196.

This command can halt ongoing SnapVault transfers from primary to secondary storage (invoked by the `snapvault start` or `snapvault update` commands or scheduled through the `snapvault snap sched -x` command), or from secondary back to primary storage (invoked by the `snapvault restore` command).

You can enter the `snapvault abort` command at the filer primary storage system or at the secondary storage system.

snapvault abort command usage

The `snapvault abort` command line can be used in the following ways:

Aborting primary to secondary storage transfers: The `snapvault abort` command can halt ongoing SnapVault transfers from primary to secondary storage that were invoked by the `snapvault start` or `snapvault update` commands, or that were scheduled through the `snapvault snap sched -x` command. At the filer console of either the primary or secondary storage system, enter the following command:

```
snapvault abort [sec_filer:]/vol/volx/sec_qtree
```

In the preceding command line:

sec_filer is the name of the SnapVault secondary storage system.

sec_qtree is the name of the secondary qtree to which the data is being transferred through SnapVault start or update commands.

Aborting secondary to primary storage transfers: The `snapvault abort` command can halt ongoing SnapVault transfers from secondary to primary storage that were invoked by the `snapvault restore` command. At the filer console of either the primary or secondary storage system, enter the following command:

snapvault abort [*prim_filer*:]/vol/*volx*/*prim_qtree*

In the preceding command line:

prim_filer is the name of the primary storage filer.

prim_qtree is the name of the primary qtree to which the data is being restored.

Aborting SnapVault snapshot creation: the `snapvault abort` command can halt the ongoing creation of a SnapVault snapshot on the secondary storage system. At the filer console of the secondary storage system, enter the following command:

snapvault abort -s *volx sv_snapname*

In the above command lines

The `-s` option specifies that the command aborts the attempt to create a snapshot with basename of *sv_snapname* on volume *volx*.

You can obtain the secondary storage system volume and snapshot basenames from the `snapvault status -s` output.

Ending SnapVault backups for a qtree

Why you end SnapVault backups

You can use the `snapvault stop` command to end the SnapVault backup process for a qtree when you no longer need the data in the primary storage system qtree to be protected.

Note

The descriptions and procedures in this section pertain to SnapVault backup of NetApp filers only. For descriptions and procedures pertaining to SnapVault backup of Open Systems drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 190 and "Managing SnapVault backup of Open Systems platforms" on page 196.

How to end a SnapVault backup

To end SnapVault backups for a specified qtree, complete the following step.

Step	Action
1	From the secondary storage system, enter the following command: **`snapvault stop [`*`sec_filer`*`:]`*`sec_qtree_path`* [*sec_filer*:]*sec_qtree_path* is the qtree that you no longer want to back up.

Example: `filerB> snapvault stop filerB:/vol/sv_vol/qtree3`

Result: SnapVault stops updating the qtree on the secondary storage system and deletes the qtree. Existing snapshots on the secondary storage system are unaffected, but as new snapshots replace the old ones, the data from the qtree whose backup was stopped will disappear.

Note

After you end the backup process from a SnapVault secondary storage system, you might want to release the now-obsolete snapshots on the primary storage system side. For more information on releasing snapshots from the primary storage system side, see "Releasing SnapVault relationships" on page 240.

Unscheduling SnapVault snapshots

Why to unschedule snapshots

You might want to unschedule a set of SnapVault snapshots if the data in the qtrees you are backing up has been migrated to another location or is no longer useful.

Note

The descriptions and procedures in this section pertain to SnapVault backup of NetApp filers only. For descriptions and procedures pertaining to SnapVault backup of Open Systems drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 190 and "Managing SnapVault backup of Open Systems platforms" on page 196.

How to unschedule snapshots

To unconfigure the SnapVault schedule for a set of snapshots and stop the snapshot process for the SnapVault primary storage system or secondary storage system, complete the following step.

Step	Action
1	From the primary or secondary storage system, enter the following command: **`snapvault snap unsched [-f] [volume [snap_name]]`** `-f` forces the command to run without showing the list of snapshots to stop creating and without asking for confirmation. *volume* is the name of the volume to stop creating snapshots on. *snap_name* is the basename of the snapshot set to stop creating. If no *snap_name* is provided, SnapVault unconfigures the SnapVault snapshot schedule for all snapshot sets in the volume and does not update any more SnapVault snapshots in the volume. If no volume is provided, SnapVault unschedules all SnapVault snapshot sets for the filer and does not update any more SnapVault snapshots in the filer. If there is already a snapshot being created in the volume, the command fails.

Example: `filerB> snapvault snap unsched vol1 sv_nightly`

Result: SnapVault asks for confirmation. If you confirm the action, SnapVault unschedules all SnapVault snapshots with the basename sv_nightly on vol1 of filerB.

Releasing SnapVault relationships

Why to release SnapVault relationships

You release a SnapVault relationship between a primary qtree and its secondary qtree backup (originally defined through the `snapvault start` command) after the relationship is no longer needed, for example:

- On a primary storage system as a part of shutting down a SnapVault relationship after a `snapvault stop` command was completed on the secondary storage system
- On the secondary storage system after data is restored to a primary storage system and you do not want to reactivate the backup relationship between the primary and secondary qtrees

Note

The descriptions and procedures in this section pertain to SnapVault backup of NetApp filers only. For descriptions and procedures pertaining to SnapVault backup of Open Systems drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 190 and "Managing SnapVault backup of Open Systems platforms" on page 196.

How to release SnapVault relationships

To release snapshots and free up the space they used, complete the following step.

Step	Action
1	Enter one of the following commands. ◆ On a primary storage system filer console, enter: **snapvault release *prim_qtree_path sec_filer:sec_qtree_path*** ◆ On the secondary storage system filer console, enter: **snapvault release *sec_qtree_path prim_filer:prim_qtree_path***

Example:

```
filerB> snapvault release /vol/sv_vol/qtree3
filerA:/vol/vol1/qtree3
```

Protecting your SnapVault backups through SnapMirror

About SnapVault secondary storage system protection

By setting up a SnapMirror relationship between the SnapVault secondary storage system and a SnapMirror destination filer, R100, or tape backup unit, you can provide backup and standby service or backup and restore protection for the SnapVault secondary storage system data.

- SnapMirror backup and standby service for SnapVault means using the SnapMirror destination device as a standby device to be activated as an alternate SnapVault secondary storage if the original secondary storage device goes down. For details see "Backup and standby service for SnapVault" on page 241.
- SnapMirror backup and restore protection for SnapVault means using the SnapMirror destination device as a source from which you can restore backup data to a SnapVault secondary storage system that has suffered data loss or corruption. For details see "Backup and restore protection for SnapVault" on page 242.

The process of SnapMirroring SnapVault: The SnapVault secondary storage system will carry out SnapVault operations on its filer sources as usual, then on a scheduled per-volume basis, the system will mirror its vaulted snapshot data to its SnapMirror destination partner or tape backup unit.

Under this configuration, SnapVault does not delete any snapshot version on the primary storage systems until that version has been successfully mirrored from the SnapVault secondary storage unit to its SnapMirror destination. This guarantees that a snapshot version will always be retrievable even if the SnapVault secondary storage system is disabled.

Backup and standby service for SnapVault

To set up SnapMirror backup and standby protection for the SnapVault secondary storage system:

1. Use the `license` command to confirm that the SnapVault secondary storage device has both SnapVault secondary storage and a SnapMirror features licensed.
2. Use the `license` command to confirm that the SnapMirror destination device has both the SnapVault secondary storage and SnapMirror features licensed.

3. Set up SnapMirror replication from the active SnapVault secondary storage system to a destination disk-based device (another filer or R100).

   For a quick description of setting up a SnapMirror relationship with a destination disk-based device, see "Setting up a basic SnapMirror operation" on page 86.

4. If the active SnapVault secondary storage system is damaged or destroyed, convert the SnapMirror destination device to an alternate SnapVault secondary system to carry on the task of backing up data from the primary storage filers.

   For details on converting a SnapMirror destination to a writable volume, see "Converting a destination to a writable volume or qtree" on page 139.

5. Activate the SnapVault license on the new SnapVault systems, and use the `snapvault start` and `snapvault snap sched` commands to complete configuration of the new SnapVault secondary storage system.

Backup and restore protection for SnapVault

To set up SnapMirror backup and restore protection for the SnapVault secondary storage system:

1. Use the `license` command to confirm that the SnapVault secondary storage device has both SnapVault secondary storage and SnapMirror features licensed.

2. If the SnapMirror destination device is a filer or R100, use the `license` command to confirm that its SnapMirror feature is licensed.

3. Set up SnapMirror replication from the active SnapVault secondary storage system to either a disk-based device or a tape device.
   - For a quick description of setting up a SnapMirror relationship with a destination disk-based device, see "Setting up a basic SnapMirror operation" on page 86.
   - For a description of setting up a SnapMirror relationship with a destination tape device, see "SnapVault-to-tape backup scenario" on page 162.

4. If the active SnapVault secondary storage system suffers data loss or corruption, use the appropriate SnapMirror commands to restore the necessary data from the SnapMirror destination device back to SnapVault secondary storage.
   - For a description of restoring to a SnapMirror source from a destination (the SnapVault secondary storage unit would be the source, and the

partner device would be the destination) see “Disaster recovery: a special use for snapmirror resync” on page 149.

- For a description of restoring to a SnapMirror source from a tape drive (the SnapVault secondary storage unit would be the source, and the tape device would be the destination) see “Restoring to SnapVault from a local tape” on page 164.

Turning SnapVault off

Why you turn SnapVault off

You might want to turn SnapVault off on a filer because the files there are no longer important or current or have been moved to another location. When you do not want SnapVault to run on a filer, you can turn it off using the `options snapvault.enable off` command.

Note

The descriptions and procedures in this section pertain to SnapVault backup of NetApp filers only. For descriptions and procedures pertaining to SnapVault backup of Open Systems drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 190 and "Managing SnapVault backup of Open Systems platforms" on page 196.

How to turn SnapVault off

To turn off SnapVault on a filer primary storage system or on a secondary storage system, complete the following step.

Step	Action
1	On both the primary storage system and the secondary storage system, enter the following command: **`options snapvault.enable off`** This option persists across reboots.

Volume Copy 6

About this chapter

This chapter discusses using the `vol copy` set of commands to copy all of the data from one volume to another. It also discusses how to perform a volume copy. This chapter does not discuss using the SnapMirror option, which automatically maintains copies (replicas) of data in one volume in another volume. For information about replicating a volume using SnapMirror, see "Data Protection Using SnapMirror" on page 71.

Topics of this chapter

This chapter discusses the following topics:

- "Learning about volume copy" on page 246
- "Preparing to copy a volume" on page 248
- "Copying volumes" on page 255
- "Checking the status of a volume copy operation" on page 260
- "Displaying the current speed for copying a volume" on page 262
- "Controlling the speed of a volume copy operation" on page 263
- "Aborting a volume copy operation" on page 265

Learning about volume copy

What a volume copy is

Volume copy is a way of copying both data in the active file system and data in snapshots from one volume to another. You can initiate a volume copy with the `vol copy start` command, which enables you to copy data from one volume to another volume, either to the same filer or a different filer. The result of copying a volume is a restricted volume containing the same data as the source volume at the time you initiated the copy operation.

Benefits of the vol copy command set

Although you can copy data on the filer using client programs such as `cpio` or use the Data ONTAP `dump` and `restore` commands, the `vol copy` command set offers the following benefits:

- When a `vol copy` command reads and writes data, Data ONTAP does not traverse directories on the filer. Data is copied block for block directly from the disks, which means that Data ONTAP can finish the copying faster than it could with other methods.
- Using a `vol copy` command, Data ONTAP preserves the snapshot data of the source volume. If, in the future, users might need to use snapshots that were taken before data was copied from one volume to another, you can use a `vol copy` command for migrating data. For example, if users accidentally delete files and need to recover them, they can do so from the preserved data.

When to copy volumes

The following table describes some situations where you might find copying volumes useful.

Situation	Reasons for copying one volume to another
You want to migrate data from one filer to another.	The destination filer has more storage or is a model that supports newer technology.

Situation	Reasons for copying one volume to another
You want to move a volume from one set of disks to another on the same filer.	You want to ◆ Split a volume ◆ Expand filer storage **Examples:** You can copy the vol0 volume to the vol1 volume and then delete duplicated files and directories in these volumes so that the original contents of vol0 are split into two volumes. You have six 9-GB disks for the vol0 volume and four 18-GB spare disks. You can migrate vol0 to the four 18-GB disks and replace all the 9-GB disks with larger capacity disks.
You want to copy data from one filer to another regularly to ensure high data availability.	After you copy the data, clients can switch to the destination filer in the following scenarios: ◆ When you shut down the source filer for software or hardware upgrades, or if the source filer is not available for reasons such as natural disasters, you can put the destination volume online to continue file service. ◆ If a network client process accidentally deletes a large number of files on the source filer, clients can continue to have access to the files on the destination filer while you are restoring the files to the source filer. **Note** This is also a good application for SnapMirror.

Preparing to copy a volume

Requirements for copying a volume

The filers involved in a volume copy operation must meet several requirements. The following list provides a brief description of these requirements. The rest of this section provides more detailed information about verifying whether the source and destination volumes meet these requirements.

- The capacity of the destination volume must be greater than or equal to the capacity of the source volume.
- The source and destination filers must have a trust relationship with each other.
- The destination volume must exist, and must not be the root volume.
- The source volume must be online and the destination volume must be restricted.
- Remote Shell access must be enabled.
- The destination volume must not contain data that you want to preserve.

Take care not to overwrite data that you need

If the destination volume is not a new volume, make sure that it does not contain data that you might need in the future. After Data ONTAP starts copying the source volume, it overwrites the entire destination volume. All data in the active file system and in the snapshots of the destination volume is lost after Data ONTAP starts copying the data.

Where volume copies can reside

The source and destination volumes of the copy can reside on the same filer or on different filers.

Recommendation for copying a volume

When a filer copies data between two volumes on separate filers, it floods the network between the two filers with packets. Users of the filers involved in a volume copy operation might notice a degradation in response time during the copy. To avoid network-related performance problems when copying to a different filer, Network Appliance recommends that you set up a private network for copying between the source and destination filers.

For detailed information

The following sections discuss the ways you prepare for copying a volume:

- "Verifying the size of each volume" on page 250
- "Verifying the relationship between filers" on page 251
- "Verifying and changing status of source and destination volumes" on page 252
- "Enabling remote access" on page 254

Verifying the size of each volume

Verifying the volume size

To see whether the data in one volume can be copied or replicated to another volume, you need to compare the file system size of the two volumes. To compare the file system size of the volumes, complete the following steps.

Step	Action
1	Enter the following command on the source filer: **`vol status -b volume_name`** *volume_name* is the name of the source volume. **Result:** The command displays the block size of the volume (in bytes), the RAID volume size, and the WAFL file system size. If no volume name is given, information for all volumes is displayed.
2	On the destination filer, repeat Step 1, replacing *volume_name* with the name of the destination volume.
3	Compare the file system (FS) numbers. If the file system size of the destination is the same as or larger than the file system size of the source, you can use the `vol copy` command (or SnapMirror) to transfer data from the source to the destination.

Example:

```
vol status -b

Volume    Block Size (bytes)   Vol Size (blocks)   FS Size (blocks)
------    ------------------   -----------------   ---------------
sourcevol 4096                 4346752             4346752
destvol   4096                 4346752             4346752
```

Verifying the relationship between filers

Creating trust relationships between filers

If the source and destination volumes in a volume copy operation reside on two different filers, the filers must have a trust relationship with each other.

To specify each filer as a trusted host of the other, complete the following steps.

Step	Action
1	By using FilerView or mounting the filer with NFS, enter the destination filer host name in the /etc/hosts.equiv file of the source filer, if it is not present already. The /etc/hosts.equiv file contains a list of host names, each of which is on a separate line. The presence of a host name in this file indicates that the filer allows that host to perform remote operations.
2	Repeat Step 1 on the destination filer, entering the source filer host name in the /etc/hosts.equiv file, if it is not present already.

Verifying and changing status of source and destination volumes

Verifying and changing volume status

To verify that the source volume is online and that the destination volume exists and is restricted, and to change the status of a volume when necessary, complete the following steps.

Step	Action
1	To verify that the destination volume exists and is restricted, enter the following command on the destination filer: **`vol status dest_volume`** *dest_volume* is the name of the volume whose status you want to check. If you do not provide a volume name, the command displays the status of all volumes in the filer. If the volume does not exist, Data ONTAP returns an error. See the *System Administrator's Guide* for information about how to create a volume. **Note** The destination volume cannot be the root volume. This is because the destination volume must be offline when Data ONTAP executes the `vol copy` command, and a root volume must always be online.
2	To verify that the source volume is online, repeat Step 1on the source filer, replacing *dest_volume* with the name of the source volume.
3	If you need to change the status of a volume because of the results of Step 1, enter the following command on the destination filer: **`vol restrict dest_volume`** *dest_volume* is the name of the destination volume.

Step	Action
4	If you need to change the status of a volume because of the results of Step 2, enter the following command on the source filer: **`vol online source_volume`** *source_volume* is the name of the source volume.
5	If you needed to perform Step 3 or Step 4, you might want to perform Step 1 or Step 2 again to verify the changes that you made.

Example:

```
filerA> vol status

        Volume    State       Status     Options
          vol0    online      normal     root
          vol1    online      normal     raidsize=14
          vol2    online      restricted
     volextra     offline
```

Enabling remote access

Enabling Remote Shell services

To perform a volume copy from one volume to another volume on the same filer, Remote Shell services must be enabled or the volume copy fails.

To enable Remote Shell services, complete the following step.

Step	Action
1	Enter the following command: `options rsh.enable on`

Copying volumes

Command to use to copy volumes

You use the `vol copy start` command to generate volume copy operations, which produce screen messages that show the progress of the operations.

Each `vol copy start` command generates two volume copy operations, each of which is assigned a number:

- One operation is for reading data from the source volume. Screen messages displayed by a `vol copy` command refer to this operation as the `volcopy dump` operation.
- One operation is for writing data to the destination volume. Screen messages displayed by a `vol copy` command refer to this operation as the `volcopy restore` operation.

When to use the volume copy operation number

You need the volume copy operation number if you want to stop a volume copy operation or change the volume copy operation speed.

For information about obtaining the volume copy operation number, see "Checking the status of a volume copy operation" on page 260.

Number of vol copy operations supported

Each filer supports a given number of simultaneous volume copy operations. For the number of copy operations supported per filer type, see "Maximum number of simultaneous replications on a filer" on page 77.

Whether Data ONTAP can execute a `vol copy start` command depends on how many volume copy operations are already in progress on the filers specified in the `vol copy start` command, as illustrated in the following examples.

Example: To copy volumes locally, you can enter the following two `vol copy start` commands on an F85 filer, which supports four simultaneous copy operations:

```
vol copy start vol0 vol1
vol copy start vol2 vol3
```

When these commands are in progress, if you enter additional `vol copy start` commands, they will fail, because four volume copy operations are already running on the filer. Two of the operations are for reading the vol0 and vol2 volumes, and two of the operations are for writing the vol1 and vol3 volumes.

Example: Suppose you enter the following three `vol copy start` commands on an F85 filer named filerA to copy volumes to another F85 filer named filerB:

```
vol copy start vol0 filerB:vol0
vol copy start vol1 filerB:vol1
vol copy start vol2 filerB:vol2
```

When these commands are in progress, filerA runs three volume copy operations to read the volumes, and filerB runs three volume copy operations to write the volumes.

An additional `vol copy start` command to copy *between* filerA and filerB will succeed because the command adds one more volume copy operation to each filer.

However, if you enter an additional `vol copy start` command to copy volumes *locally* on either filerA or filerB, it will fail. This is because the additional command creates two volume copy operations, one for reading and one for writing, on the filer that performs the local copying.

Copying snapshots with the vol copy start command

The following table describes the snapshots that will be copied from the source volume and the resulting snapshots on the destination volume, depending on the option you use with the `vol copy start` command.

Option	Snapshots to copy from the source volume	Snapshots in the snapshot file system of the destination volume
None	No snapshots in the file system will be copied. Only the snapshot taken after you enter the `vol copy start` command will be copied.	A snapshot named snapshot_for_volcopy.*n*, where *n* is a number starting at 0 and incrementing by one whole number with each `vol copy` operation

Option	Snapshots to copy from the source volume	Snapshots in the snapshot file system of the destination volume
`-S`	All snapshots in the snapshot file system of the source volume and the snapshot taken after you enter the `vol copy start command` will be copied.	All snapshots in the source volume and snapshot_for_volcopy.*n*, where *n* is a number starting at 0 and incrementing by one whole number with each vol copy operation.
`-s` followed by the name of the snapshot	The specified snapshot will be copied.	The specified snapshot

Note

The `vol copy start -S` command does not copy any snapshots that are created while the copying is in progress. For example, if the copying lasts from 11:45 p.m. to 1:00 a.m. the next day and Data ONTAP creates a snapshot named nightly.0 at midnight, Data ONTAP does not copy the nightly.0 snapshot.

Copying one volume to another

To copy one volume to another, complete the following step.

Step	Action
1	Enter the following command: **`vol copy start [-S | -s snapshot_name] source_volume dest_volume`** The `-S` and `-s` arguments specify the snapshots to copy. *source_volume* and *dest_volume* are the names of the source and destination volumes. If a volume is on a different filer, precede the volume name with the filer name and a colon. For examples illustrating how to specify volume names, see "Examples of the vol copy start command" in the following table. **Note** If the copying takes place between two filers, you can enter the `vol copy start` command on either the source or destination filer. You cannot, however, enter the command on a third filer that does not contain the source or destination volume.

Examples of the vol copy start command

The following table shows several examples of the `vol copy start` command.

If you want to...	Use...
Copy all snapshots from the vol0 volume to the vol1 volume on the same filer	`vol copy start -S vol0 vol1`
Copy a nightly snapshot from the vol0 volume to the vol1 volume on the same filer	`vol copy start -s nightly.1 vol0 vol1`
Create a snapshot in the vol0 volume to be copied to the vol1 volume on the same filer	`vol copy start vol0 vol1`

If you want to...	Use...
Copy all snapshots from the vol0 volume to the vol1 volume on a different filer named filerA	`vol copy start -S vol0 filerA:vol1`

Error messages generated by vol copy start commands

If your filer does not meet the requirements for copying a volume, the `vol copy start` command generates one or more error messages. The following table explains the possible error messages and their meanings.

Error message	Meaning

`Permission denied.`
 	
`VOLCOPY: Could not connect to filer filerB`
	The source filer does not have permission to copy to the destination filer.
	
	Action: Make sure that the filers have a trust relationship with each other.

`VOLCOPY: volcopy restore: volume is online, aborting`	The destination volume is online. **Action:** Take the destination volume offline.
`VOLCOPY: volcopy restore: volume is too small, aborting`	The destination volume is smaller than the source volume. **Action:** Add more disk space to the destination volume or choose another destination volume of sufficient capacity.
`write: setting up STDERR broken pipe`	A local volume copy tried to start, but Remote Shell access is not enabled on the filer. **Action:** Enable Remote Shell access on the filer so that it can receive `rsh` commands.

Checking the status of a volume copy operation

Command to use to check status

You use the `vol copy status` command to check the status of volume copy operations.

This command displays the status for a specified volume copy operation. If you do not specify the operation number, the command displays the status of all volume copy operations in progress. In the command output, the operations are differentiated from one another with unique volume copy operation numbers.

Restrictions

Keep the following restrictions in mind when checking volume copy status:

- If you start a volume copy operation from the filer console, you can enter the `vol copy status` command only through the `rsh` command when the copy operation is in progress. This is because you do not have access to the filer prompt on the console when Data ONTAP is copying the volume.
- If data is being copied between two filers, you can enter the `vol copy status` command through a Remote Shell connection to either filer. The operation numbers displayed on the source filer and the destination filer are different because the reading and the writing are considered two different operations.

Checking operation status

To check the status of a volume copy operation in progress, complete the following step.

Step	Action
1	Enter the following command: **`vol copy status [`*`operation_number`*`]`** *operation_number* is the specific volume copy operation. Omit *operation_number* to display the status of all current volume copy operations. The operations are numbered from 0 through 3.

Sample status message from vol copy start command

The following example shows a `vol copy start` command that copies the vol0 volume to the vol1 volume on the same filer. When the operation is in progress, it displays the volume copy operation status.

```
filerA>vol copy start -S vol0 vol1
Copy Volume: vol0 on machine 127.0.0.1 to Volume: vol1
Reading the dump stream
VOLCOPY: Starting on volume 1.
This dump contains 257 blocks
10:04 pm : volcopy restore 1 : begun.
10:04 pm : volcopy restore 1 : 5 % done. Estimate 3 minutes
remaining.
.
.
.
10:04 pm : volcopy restore 1 : 95% done. Estimate 1 minutes
remaining.
```

Example of the vol copy status command using rsh

Before the prompt is displayed again, you can use the `vol copy status` command on a trusted host of the filer, as shown in the following example:

```
rsh filerA vol copy status
10:04 pm : volcopy dump 0 : 99 % done. Estimate 1 minutes remaining.
10:04 pm : volcopy restore 1 : 99 % done. Estimate 1 minutes
remaining.
No operation 2 in progress.
No operation 3 in progress.
```

In the previous examples, volume copy operation 0, shown as `volcopy dump 0` in the display, is for reading the data from the vol0 volume; volume copy operation 1, shown as `volcopy restore 1` in the display, is for writing the data to the vol1 volume.

Displaying the current speed for copying a volume

When to display the volume copy speed

You can display the speed for copying a volume when you want to determine the current setting, and to verify the speed before changing the setting. This procedure enables you to verify the default speed for all volume copy operations.

Displaying the default speed for copying a volume

To display the speed for copying a volume, complete the following step.

Step	Action
1	Enter the following command: `options vol.copy.throttle` **Result:** The value 10 (full speed) through 1 (one-tenth full speed) to be used by all volume copy operations is displayed. The default value is 10.

Controlling the speed of a volume copy operation

When to control volume copy speed

You might want to control the speed of a volume copy operation at two times:

- Before you start the volume copy operation
- During a volume copy operation

Note

The speed for reading data from the source volume and the speed for writing data to the destination volume can be different. The smaller of the two values determines the time required for Data ONTAP to finish copying the data.

Why you change volume copy speed

You can change the speed of a volume copy operation when you suspect it might cause performance problems on your filer.

Note

Changing the `vol.copy.throttle` option changes the default speed for all volume copy operations to follow.

Controlling volume copy operation speed

To control volume copy operation speed, complete the following step.

Step	Action	
1	**If you want to control the speed of the volume copy...**	**Then...**
	Before you start the copy operations	Enter the following command: **`options vol.copy.throttle value`** *value* is the specific speed you want.

Step	Action	
	During the copy operation	Enter the following command through a Remote Shell: **vol copy throttle [*operation_number*] *value*** *operation_number* is the specific volume copy operation whose speed you want to adjust. If you do not specify an operation number, the command applies to all volume copy operations that are in progress. *value* is the specific speed you want.

Example

The following example illustrates changing the speed of all volume copy operations in progress to one-tenth of full speed through a Remote Shell:

```
rsh filerA vol copy throttle 1
volcopy operation 0: Throttle adjusted from 100% to 10%.
volcopy operation 1: Throttle adjusted from 100% to 10%.
```

Aborting a volume copy operation

About aborting a volume copy

If data is being copied between two filers, you can execute the `vol copy abort` command on either filer.

If you start the volume copying operation from the filer console, you can enter the `vol copy abort` command only through the `rsh` command. This is because you do not have access to the filer prompt on the console during the copying.

Caution

An incomplete volume copy operation leaves unusable data in the destination volume.

Prerequisite

To abort a specific volume copy operation, you need to specify the volume copy operation number. You can obtain the operation number from the `vol copy status` output.

Aborting a volume copy

To abort a volume copy operation, complete the following step.

Step	Action
1	Enter the following command: **`vol copy abort [all | operation_number]`** *operation_number* is the specific volume copy operation to be aborted. Specify `all` to abort all operations.

Mirrored Volume Management Using SyncMirror 7

About this chapter

This chapter describes how to use the optional SyncMirror™ feature of Data ONTAP to mirror volumes and how to manage them.

Topics in this chapter

This chapter discusses the following topics:

- "Understanding mirrored volumes" on page 268
- "Enabling and disabling the mirroring license" on page 271
- "Creating mirrored volumes" on page 272
- "Adding disks to mirrored volumes" on page 282
- "Changing the state of a plex" on page 286
- "Splitting mirrored volumes" on page 288
- "Removing and destroying a plex from a mirrored volume" on page 290
- "Comparing the plexes of mirrored volumes" on page 291

Understanding mirrored volumes

About sync-mirrored volumes

The SyncMirror software creates volumes that consist of two copies of the same file system. The two copies, known as plexes, are simultaneously updated; therefore, the copies are always identical.

Advantage of SyncMirror

Sync-mirrored volumes provide a high level of data availability by physically separating the two plexes of a sync-mirrored volume. Additionally, each plex has its own collection of spare disks that are separated from the other plex‘s spare disks.

This physical separation of the plexes protects against loss of data if a double-disk error or loss of disk connectivity occurs. The unaffected plex continues to serve data while you fix the cause of the failure. Once fixed, the two plexes can be resynchronized and the mirrored relationship reestablished. Another advantage of mirrored plexes is faster rebuild time.

In contrast, if a SnapMirrored volume goes down, its SnapMirror partner cannot automatically take over the file serving functions and can only restore data to its condition at the time of the last snapshot creation.

Disadvantage of SyncMirror

The disadvantage of SyncMirror is that you must use twice as many disks on a mirrored volume as you would on an unmirrored volume. Each of the two plexes requires a full set of disks. For example, you need 200 GB of disk space to mirror a 100 GB volume, 100 GB for each plex of the mirrored volume.

Prerequisites for using mirrored volumes

The following are prerequisites for using mirrored volumes:

- You must purchase and enable a SyncMirror license.
- You must have an F800 series filer or F800 series cluster.
- You must have DS14 or FC9 disk shelves.
- You must connect disk shelves to the filer in a configuration that supports mirrored volumes. For more information about connecting disk shelves to support mirrored volumes, see the *System Configuration Guide*, *F800 Series Cluster Guide*, *Fibre Channel DiskShelf14 Hardware Guide*, and *Fibre Channel StorageShelf FC9 Hardware Guide*.

Note

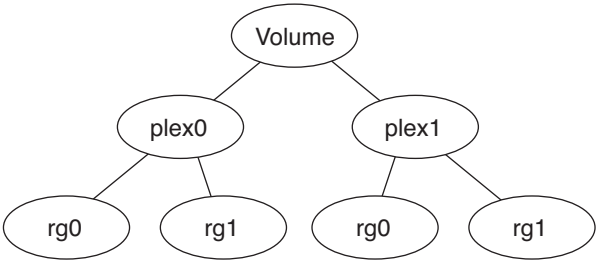
The SyncMirror function is not supported on cluster configurations that use the Servernet cluster card. The SyncMirror function is supported on clusters using a Troika cluster card.

What mirrored volumes are

A mirrored volume is a single volume with two physically separated and synchronously up-to-date copies on disks.

What a file system is called

The two copies of a mirrored volume are called plexes. Each plex is a physical copy of the same file system and consists of one or more RAID groups. The following diagram illustrates the concept of a plex in a mirrored volume.



Plex naming convention

The convention used by Data ONTAP to name plexes is as follows:

vol-name/plex*plexid*

vol-name is the name of the volume.

plexid is the number of the plex and is assigned by Data ONTAP.

Example: The mirrored volume vol0 has two plexes, vol0/plex0 and vol0/plex1.

Where mirrored volume plexes get their disks

When a mirrored volume is created, Data ONTAP takes spare disks from a collection of disks, called a disk pool, to create one plex and spare disks from another disk pool to create the other plex.

Note

To enable a filer for mirrored volumes, you must configure the filer and disk shelves to use two disk pools.

How disk pools are created

When you configure your filer to use mirrored volumes, you connect the shelves to the filer in a way that physically separates spare disks into two pools. You do this by using separate cables and adapters to connect separate shelves to the filer.

After you configure your filer and disk shelves and enable the mirrored volume feature, Data ONTAP divides spare disks into two different logical collections called disk pools. When Data ONTAP creates disk pools, it uses your configuration to ensure that one disk pool is physically separate from the other disk pool. Disk pools must be physically separate from one another to ensure high availability of the mirrored volume.

Advantage of mirrored volumes

The advantage of mirrored volumes over unmirrored volumes is that there is a higher availability of data. Mirrored data exists on separate shelves connected to the filer by separate cables and adapters to avoid single points of failure. Because of this separation and because the two plexes of a mirrored volume contain the same data at the same time, one plex can fail and you will still have access to the data through the other plex.

Enabling and disabling the mirroring license

Enabling the mirroring license

The license code name for the mirroring feature is localsyncmirror. To enable the mirroring license, complete the following steps.

Step	Action
1	Enter the following command: **`license add xxxxxxx`** *xxxxxxx* is the localsyncmirror license code you purchased.
2	Reboot the filer.

Disabling the mirroring license

You cannot disable the mirroring license if one or more mirrored volumes exist and are online.

To disable the mirroring license, complete the following steps.

Step	Action
1	Take all mirrored volumes offline by entering the following command: **`vol offline vol-name`** *vol-name* is the name of the mirrored volume **Note** Only one volume at a time can be taken offline.
2	Repeat Step 1 until all mirrored volumes are offline.
3	Enter the following command: **`license delete localsyncmirror`**

Creating mirrored volumes

Two ways to create mirrored volumes

You can create a mirrored volume in the following ways:

- You can create a new volume that has two plexes.
- You can add a plex to an existing unmirrored volume.

Note

A mirrored volume cannot have more than two plexes.

How Data ONTAP selects disks

Regardless of which way you create a mirrored volume, when you create a mirrored volume or add disks to a mirrored volume, Data ONTAP determines what disks to use. Data ONTAP uses the following disk selection policies when selecting disks for mirrored volumes:

- Disks selected for each plex must come from different disk pools.
- The number of disks selected for one plex must equal the number of disks selected for the other plex.
- Disks are selected first on the basis of equivalent bytes per sector (bps) size, then on the basis of the size of the disk.
- If there is no equivalent-sized disk, Data ONTAP takes a larger capacity disk and downsizes it.

Disk selection policies if you select disks

Data ONTAP allows you to select disks when creating mirrored volumes or adding disks to a mirrored volume. Follow the same disk selection policies that Data ONTAP follows when selecting disks for mirrored volumes. See the previous section, "How Data ONTAP selects disks", for a list of disk selection policies.

See "Creating a mirrored volume when creating a volume" on page 275 and "Adding a plex to an existing unmirrored volume" on page 279 for information about creating mirrored volumes with disks you select.

Data ONTAP names plexes

Regardless of how a mirrored volume is created, Data ONTAP names the plexes of the mirrored volume. For more information about the plex naming convention, see "Plex naming convention" on page 269.

Viewing plex and disk pools

Viewing the disks in a plex or the spare disks in a disk pool is useful when adding disks to a plex or to understand which plex is using which disk pool. To view plexes and disk pools, complete the following step.

Step	Action
1	Enter one of the following commands: **sysconfig -r** or **vol status -r**

Example: In this example, the `vol status -r` command is used to view the disks in plexes and spare disks in disk pools:

```
filer2> vol status -r
Volume vol0 (online, normal) (block checksums)
  Plex /vol0/plex0 (online, normal, active)
    RAID group /vol0/plex0/rg0 (normal)

RAID Disk Device HA  SHELF BAY CHAN  Used (MB/blks) Phys (MB/blks)
--------- ------ ------------- ---- ---------------- ----------------
parity     6.0   6    0    0   FC:A  17000/34816000 17160/35145032
data       6.2   6    0    2   FC:A  17000/34816000 17160/35145032

Volume newVol (online, mirrored) (block checksums)
  Plex /newVol/plex0 (online, normal, active)
    RAID group /newVol/plex0/rg0 (normal)

RAID Disk Device HA  SHELF BAY CHAN  Used (MB/blks) Phys (MB/blks)
--------- ------ ------------- ---- ---------------- ----------------
parity     6.1   6    0    1   FC:A  17000/34816000 17160/35145032
data       6.3   6    0    3   FC:A  17000/34816000 17160/35145032

  Plex /newVol/plex2 (online, normal, active)
    RAID group /newVol/plex2/rg0 (normal)

RAID Disk Device HA  SHELF BAY CHAN  Used (MB/blks) Phys (MB/blks)
--------- ------ ------------- ---- ---------------- ----------------
parity     3.17  3    2    1   FC:A  17000/34816000 17160/35145032
data       3.16  3    2    0   FC:A  17000/34816000 17160/35145032
```

```
Pool1 spare disks
RAID Disk Device HA  SHELF BAY CHAN  Used (MB/blks) Phys (MB/blks)
---------------------------------------------------------------------
Spare disks for block or zoned checksum volumes
spare      3.18  3   2     2    FC:A  34000/69632000 34190/70022840
spare      3.19  3   2     3    FC:A  34000/69632000 34190/70022840
spare      3.20  3   2     4    FC:A  34000/69632000 34190/70022840
spare      3.21  3   2     5    FC:A  34000/69632000 34190/70022840
spare      3.22  3   2     6    FC:A  34000/69632000 34190/70022840

Pool0 spare disks

RAID Disk Device HA  SHELF BAY CHAN  Used (MB/blks) Phys (MB/blks)
---------------------------------------------------------------------
Spare disks for block or zoned checksum volumes
spare      6.18  6   0     4    FC:A  17000/34816000 17160/35145032
spare      6.19  6   0     5    FC:A  17000/34816000 17160/35145032
spare      6.20  6   0     6    FC:A  17000/34816000 17160/35145032
spare      6.21  6   1     0    FC:A  17000/34816000 17160/35145032
spare      6.22  6   1     1    FC:A  17000/34816000 17160/35145032
```

For detailed information

The following sections discuss the ways you can create mirrored volumes:

- "Creating a mirrored volume when creating a volume" on page 275
- "Adding a plex to an existing unmirrored volume" on page 279

Creating a mirrored volume when creating a volume

Methods for creating a mirrored volume

You can create a mirrored volume when creating a volume using one of the following methods:

- Let Data ONTAP select the disks used to mirror the volume.
- Select the disks used to mirror the volume yourself.
- Preview the disks Data ONTAP would use, then either use those disks or replace one or more of the disks with disks you choose.

Data ONTAP selects the disks

To create a mirrored volume by letting Data ONTAP select the disks used, complete the following step.

Step	Action
1	At the prompt, enter the following command: **`vol create vol-name -m ndisks[@disk-size]`** *vol-name* is the name for the new volume (without the /vol/ prefix). `-m` specifies a mirrored volume to Data ONTAP. *ndisks* is the number of disks to use; *ndisks* must be at least 4 and an even number. Even numbers are required because half of the disks specified by *ndisks* go to one plex from one disk pool and half go to the other plex from another disk pool. *disk-size* is a number that specifies the disk size to use, in gigabytes. You must have half of the *ndisks* available disks of the size you specify in one disk pool and half of the *ndisks* available disks of the size you specify in another disk pool if you use this option. **Note** If you use the `disk-size` option, first determine the size of spare disks in the disk pools using the `vol status -r` command. **Example:** In this example, use the following command to create a mirrored volume that has two plexes, each plex containing three disks: `vol create volA -m 6`

You select the disks

If you select the disks used to create a mirrored volume, you must ensure that you use the correct number of disks, the correct size of disks, and that the disks for each volume plex come from different pools.

To create a mirrored volume using disks you select, complete the following step.

Step	Action
1	At the prompt, enter the following command: **`vol create vol-name -m -d disk-list -d disk-list`** `-m` specifies a mirrored volume to Data ONTAP. *vol-name* is the name of the volume you are mirroring (without the /vol/ prefix). *disk-list* consists of the disk IDs of two or more available disks; separate multiple disks with a space. Both `-d` options must be used, one for each plex. **Note** Specifying only one disk set using the `-d` option will fail. **Example:** In this example, use the following command to create a mirrored volume named vol A with disks 6.1 and 6.2 on one plex and with disks 8.1 and 8.2 on the other plex: `vol create volA -m -d 6.1 6.2 -d 8.1 8.2`

You select the disks with assistance from Data ONTAP

Data ONTAP has a preview option that you can use to view what disks Data ONTAP would select if it were to create a mirrored volume.

To create a mirrored volume by previewing disks Data ONTAP would use in the form of a sample command, complete the following steps.

Step	Action	
1	At the prompt, enter the following command: **`vol create vol-name -n -m ndisks[@disk-size]`** *vol-name* is the name of the mirrored volume you are creating (without the /vol/ prefix). `-n` specifies to Data ONTAP not to create the mirrored volume, but to show the disks that it would use if it did create the mirrored volume. `-m` specifies a mirrored volume to Data ONTAP. *ndisks* is the number of disks to use. Creating a mirrored volume will fail if the number of disks is an odd number. *disk-size* is a number that specifies the disk size to use, in gigabytes. You must have at least *ndisks* available disks of the size you specify. **Result:** Data ONTAP returns a `vol create` command that specifies the disks.	
2	**If you want to...**	**Then...**
	Use the disks specified by Data ONTAP	At the prompt, enter the `vol create` command specified.
	Substitute other disks for one or more disks specified by Data ONTAP	At the prompt, enter the `vol create` command and substitute other disks for one or more of the disks specified.

Example: In this example, use the following command to preview the command that Data ONTAP would use when creating a 4-disk, mirrored volume called volB:

```
vol create volB -n -m 4
```

Data ONTAP returns the following command:

```
vol create volB -m -d 5.1 5.3 -d 8.3 8.4
```

Adding a plex to an existing unmirrored volume

Mirroring an unmirrored volume that uses mixed disk capacities

The unmirrored volume that you want to mirror might use disks of different capacities. For example, the volume might consist of several 36-GB disks and one 72-GB disk. Data ONTAP can mirror volumes that use mixed disk capacities by selecting disks that match the smallest capacity from a different disk pool. If there are not enough disks of that capacity in the disk pool, Data ONTAP selects a higher capacity disk and downsizes it.

See "How Data ONTAP selects disks" on page 272 for the rules Data ONTAP uses to select disks.

Methods for adding a plex to an unmirrored volume

You can add a plex to an existing unmirrored volume using one of the following methods:

- Let Data ONTAP select the disks used to mirror the volume.
- Select the disks used to mirror the volume yourself.
- Preview the disks Data ONTAP would use, then either use those disks or replace one or more of the disks with disks you choose.

Data ONTAP selects the disks

If Data ONTAP selects the disks used to mirror the volume, Data ONTAP adds the necessary number and type of disks from a separate pool. This is the easiest method of mirroring, but you do not have control over which disks Data ONTAP selects.

To mirror an existing volume by letting Data ONTAP select the disks used to mirror the volume, complete the following step.

Step	Action
1	At the prompt, enter the following command: **`vol mirror vol-name`** *vol-name* is the name of the volume you are mirroring (without the /vol/ prefix).

You select the disks

If you select the disks used to mirror the volume, you must ensure that you use the correct number of disks, the correct size of disks, and that the disks are in a different pool from the volume plex that you are mirroring.

To mirror an existing volume using disks you select, complete the following step.

Step	Action
1	At the prompt, enter the following command: **`vol mirror vol-name -d disk-list`** *vol-name* is the name of the volume you are mirroring (without the /vol/ prefix). *disk-list* consists of disk IDs of one or more available disks; separate multiple disks with a space.

You select the disks with assistance from Data ONTAP

Data ONTAP has a preview option that you can use to view what disks Data ONTAP would select if it were to mirror the volume.

To mirror an existing volume by previewing disks Data ONTAP would use in the form of a sample command, complete the following steps.

Step	Action
1	At the prompt, enter the following command: **`vol mirror vol-name -n`** `-n` specifies to Data ONTAP not to create the mirrored volume, but show the disks that it would use if it did create the mirrored volume. *vol-name* is the name of the volume you are mirroring (without the /vol/ prefix). **Result:** Data ONTAP returns a `vol mirror` command that specifies the necessary number and type of disks you can use to mirror the volume.

Step	Action	
2	**If you want to...**	**Then...**
	Use the disks specified by Data ONTAP	At the prompt, enter the `vol mirror` command specified.
	Substitute other disks for one or more disks specified by Data ONTAP	At the prompt, enter the `vol mirror` command and substitute other disks for one or more of the disks specified.

Example: In this example, use the following command to preview the disks that Data ONTAP would use when mirroring the volA volume:

```
vol mirror volA -n
```

Data ONTAP returns the following command:

```
vol mirror volA -d 8.1 8.2 8.3
```

Adding disks to mirrored volumes

Rules to follow when adding disks

When you add disks to a mirrored volume, you must follow these rules:

- The number of disks you add must be an even number and equally divided between the two plexes.
- The disks that you add to each plex must come from different disk pools.
- The disks you add must have equivalent bytes per sector (bps) size.
- The disks you add must have capacity equal to or greater than those disks in the plexes.

Note

If you add a larger capacity disk, Data ONTAP downsizes the larger disk to an equivalent size of the disks in the volume.

Methods for adding disks

You can add disks to mirrored volumes using the following methods:

- Let Data ONTAP select the disks.
- Select the disks yourself.
- Preview the disks Data ONTAP would use, then either use those disks or replace one or more of the disks with ones you choose.

Data ONTAP selects the disks

To add disks to a mirrored volume, complete the following step.

Step	Action
1	At the prompt, enter the following command: **`vol add vol-name ndisks[@disk-size]`** *vol-name* is the name of the volume to which you are adding disks. *ndisks* is the number of disks to use. Adding new disks will fail if the number of disks is an odd number. *disk-size* is a number that specifies the disk size to use, in gigabytes. You must have half of the *ndisks* available disks of the size you specify in one disk pool and half of the *ndisks* available disks of the size you specify in another disk pool if you use this option. **Note** If you use the `disk-size` option, first determine the size of spare disks in the disk pools using the `vol status -r` command.

You select the disks

If you select the new disks to add to the mirrored volume, you must ensure that you use the correct number of disks, the correct size of disks, the disks have the same checksum compatibility, and that the disks you add to one volume plex are in a different pool from the disks you add to the other volume plex.

To add new disks to a mirrored volume using disks you select, complete the following steps.

Step	Action
1	Enter the following command to list available spare disks: **`vol status -r`**
2	Using the rules listed in “Rules to follow when adding disks” on page 282, choose the spare disks you want to add.

Step	Action
3	At the prompt, enter the following command: **`vol add vol-name -d disk-list -d disk-list`** *vol-name* is the name of the volume to which you are adding disks (without the /vol/ prefix). *disk-list* consists of disk IDs of one or more available disks; separate multiple disks with a space. Both `-d` options must be used, one for each plex. Data ONTAP automatically assigns the disks to the appropriate plex. **Note** Specifying only one disk set using the `-d` option will fail. **Example:** In this example, use the following command to add disk 3.1 to one plex and disk 8.1 to the other plex of the volD mirrored volume: `vol add volD -d 6.1 -d 8.1`

You select the disks with assistance from Data ONTAP

Data ONTAP has a preview option that you can use to view what disks Data ONTAP would select if it were to add new disks to the volume.

To add new disks to an existing volume by previewing disks Data ONTAP would add in the form of a sample command, complete the following steps.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td>1</td><td colspan="2">At the prompt, enter the following command:
vol add vol-name -n ndisks[@disk-size]

vol-name is the name of the mirrored volume to which you are adding disks (without the /vol/ prefix).

ndisks is the number of disks to use. Adding new disks will fail if the number of disks is an odd number.

disk-size is a number that specifies the disk size to use, in gigabytes. You must have half of the ndisks available disks of the size you specify in one disk pool and half of the ndisks available disks of the size you specify in another disk pool if you use this option.

Note
If you use the <code>disk-size</code> option, first determine the size of spare disks in the disk pools using the <code>vol status -r</code> command.

Result: Data ONTAP returns a <code>vol add</code> command that specifies the disks it would add to each plex of the mirrored volume.</td></tr>
<tr><td>2</td><td>If you want to...</td><td>Then...</td></tr>
<tr><td></td><td>Use the disks specified by Data ONTAP</td><td>At the system prompt, enter the <code>vol add</code> command specified.</td></tr>
<tr><td></td><td>Substitute other disks for one or more disks specified by Data ONTAP</td><td>At the system prompt, enter the <code>vol add</code> command and substitute other disks for one or more of the disks specified.</td></tr>
</table>

Example: In this example, use the following command to preview the disks that Data ONTAP would use when adding two new 36 GB disks to the volA mirrored volume:

```
vol add volA -n 2@36
```

Data ONTAP returns the following command:

```
vol add volA -d 6.4 -d 8.6
```

Changing the state of a plex

Plex states

A plex can be in one of the following states:

- Offline—The plex is not available for read or write access.
- Online—The plex is available for read or write access and the contents of the plex are current. An online plex can be in the following states:
  - Active—The plex is available for use.
  - Adding disks—Data ONTAP is adding disks to the RAID group or groups of the plex.
  - Empty—The plex is part of a volume that is being created and Data ONTAP needs to zero out one or more of the disks targeted to the volume before adding the disks to the plex.
  - Failed—One or more of the RAID groups in the plex failed.
  - Inactive—The plex is not available for use.
  - Normal—All RAID groups in the plex are functional.
  - Out-of-date—The plex contents are out of date and the other plex of the volume has failed.
  - Resyncing—The plex contents are being resynchronized with the contents of the other plex of the volume.

Viewing the status of plexes

To view the status of plexes, complete the following step.

Step	Action
1	Enter one of the following commands: **`sysconfig -r`** or **`vol status -r`**

Changing the plex state

Data ONTAP changes the state of a plex to resyncing when synchronizing the two plexes of a mirrored volume. For example, when you create a mirrored volume by adding a plex to an existing unmirrored volume, Data ONTAP puts the added plex in a resyncing state.

Data ONTAP allows you to change the state of a plex from offline to online and from online to offline.

To change the state of a plex, complete the following step.

Step	Action
1	Enter the following command: **`vol online | offline plexname`**

Splitting mirrored volumes

What splitting a mirrored volume is

Splitting a mirrored volume removes the mirrored data relationship between the two plexes and creates two independent unmirrored volumes. After splitting, the new volume is in a restricted state and the original volume stays in an online state.

Note

You cannot rejoin the two new unmirrored volumes to recreate the original mirrored volume. To mirror one or both of these volumes, see "Adding a plex to an existing unmirrored volume" on page 279.

Why you might split a mirrored volume

You might split a mirrored volume for one of the following reasons:

- You want to unmirror the mirrored volume. This is an alternative to the unmirroring procedure described in "Removing and destroying a plex from a mirrored volume" on page 290.
- You want to move a mirrored volume to another location.
- You want to modify the mirrored volume, but you want to test the modification before applying it. You can apply the modifications and test them on the split-off copy of the plex and then apply those changes back to the untouched original plex.

Note

You do not need to stop applications that are running before splitting a mirrored volume.

Splitting a mirrored volume

To split a mirrored volume into two independent unmirrored volumes, complete the following steps.

Note

Ensure that both plexes of the mirrored volume you are splitting are online and operational. See "Changing the state of a plex" on page 286 for information about viewing the state of a plex.

Step	Action
1	Enter the following command: **`vol split plexname volname`** *plexname* is the name of one of the plexes in the mirrored volume. *volname* is the name of the new volume. **Example:** In this example, use the following command to split plex0 from mirrored volume vol0 and name the new volume volNew: `vol split vol0/plex0 volNew` After splitting, there are two unmirrored volumes, vol0 and volNew.

Volume structure after splitting

Before splitting, a mirrored volume has two plexes, plex0 and plex1. After splitting, the new unmirrored volume with the new name has one plex, plex0. The new unmirrored volume with the original name also has one plex, but depending on which plex you chose when splitting, the plex name can be either plex0 or plex1.

The plex name for an unmirrored volume is unimportant because the volume has only one plex. If you remirror one of these unmirrored volumes, the resulting mirrored volume plex names will always be plex0 and plex1.

Removing and destroying a plex from a mirrored volume

Why you might remove and destroy a plex

You might remove a plex from a mirrored volume and destroy it if there was a problem with that plex. For example, if you have a double-disk failure that causes a plex to fail, you can remove the plex from the mirrored volume, fix the problem, and then recreate it, or you can recreate it using a different set of disks if the problem cannot be fixed.

Another reason why you might remove a plex from a mirrored volume and destroy it is if you want to unmirror the mirrored volume.

Removing and destroying a plex

To remove and destroy a plex from a mirrored volume, complete the following steps.

Step	Action
1	Ensure that the plex you are removing is offline.
2	Enter the following step: **`vol destroy plexname`** *plexname* is the name of the plex you are removing.

Example: In this example, use the following command to remove and destroy the vol0/plex1 plex from the vol0 mirrored volume after taking the vol0/plex1 plex offline:

```
vol destroy vol0/plex1
```

Result of removing and destroying a plex

Removing and destroying a plex from a mirrored volume results in an unmirrored volume because the volume now has only one plex.

After removing the plex, Data ONTAP converts the good disks used by the plex into hot spare disks.

Comparing the plexes of mirrored volumes

When you should compare plexes

Because the plexes of mirrored volumes are almost never unsynchronized, you rarely need to compare the plexes of a mirrored volume. If you do need to compare the plexes of a mirrored volume, Data ONTAP enables you to do the following tasks:

- Compare plexes

  Note
  Comparing plexes might affect filer performance.

- Stop comparing plexes
- Suspend and resume comparing plexes
- View the progress of the comparison

Comparing plexes

The mirrored volume must be online before you can compare the plexes. When comparing the two plexes of a mirrored volume, you can choose one of the following options:

- Data ONTAP compares plexes and corrects differences it finds. Network Appliance recommends this option.
- Data ONTAP compares plexes but you specify that it should not correct differences. If you choose this option, you must determine which plex is the good one and synchronize the other plex to it. The process to determine the good plex can be difficult and can use advanced Data ONTAP commands. You must consult with Network Appliance Technical Support before correcting differences using this option.

To compare the two plexes of a mirrored volume, complete the following steps.

Step	Action	
1	**If you...**	**Then...**
	Want Data ONTAP to correct differences	Enter the following command: **`vol verify start volname`** *volname* is the name of the mirrored volume whose plexes you are comparing. If no *volname* is specified, the plexes of all online mirrored volumes are compared.
	Do not want Data ONTAP to correct differences	Enter the following command: **`vol verify start volname -n`** If *volname* is not specified, Data ONTAP compares the plexes of a mirrored volume for all volumes.

Stopping plex comparison

You might need to stop Data ONTAP from comparing mirrored volume plexes. For example, you might stop verifying because it affects filer performance. To stop Data ONTAP from comparing plexes, complete the following step.

Step	Action
1	Enter the following command: **`vol verify stop volname`** If *volname* is not specified, Data ONTAP stops comparing plexes for all volumes.

Suspending and resuming plex comparison

Rather than stopping Data ONTAP from comparing the plexes of a mirrored volume, you can suspend comparing plexes. The comparison remains suspended until you resume it, stop it, or reboot the filer.

To suspend or resume a comparison of plexes, complete the following step.

Step	Action	
1	**If you want to...**	**Then...**
	Suspend the comparison	Enter the following command: **`vol verify suspend volname`** If *volname* is not specified, Data ONTAP suspends comparing plexes for all volumes.
	Resume the comparison	Enter the following command: **`vol verify resume volname`** If *volname* is not specified, Data ONTAP resumes comparing plexes for all volumes.

Viewing the status of a plex comparison

Plex comparison status tells you what percentage of the plex comparison has been completed. Plex comparison status also displays whether plex comparison of a mirrored volume is suspended.

To view the status of a plex comparison, complete the following step.

Step	Action
1	Enter the following command: **`vol verify status volname`** If *volname* is not specified, Data ONTAP shows the status of all volumes whose plexes are currently being compared.

Tape Drive Management 8

About this chapter

This chapter describes methods for using tape *devices* (software functionality designations), tape *drives* (the physical hardware modules that house and control the tapes), and tape aliases (methods of addressing tape drives). You need to manage tape drives when you back up data from the filer to tape or when you recover data from tape to the filer. In particular, you need to understand the relationship between tape devices and tape drives so you can specify devices in performing backups and restores. Tape aliases are names you assign to tape drives to make it easier to keep track of tape drives used for multiple filers.

In this and subsequent chapters, the term *tape device* means a specific functionality of a physical tape drive that you create by specifying information in the tape device name when you install a tape drive, which handles a single tape, or a tape stacker or tape library (medium changer), which handles multiple tapes.

Topics in this chapter

This chapter discusses the following topics:

- "Understanding tape devices" on page 296
- "Displaying tape device statistics" on page 299
- "Displaying supported tape drives" on page 301
- "Assigning tape aliases" on page 303
- "Displaying tape device and tape drive information" on page 311
- "Understanding tape files" on page 315
- "Controlling tape drives" on page 316
- "Qualifying tape drives dynamically" on page 321
- "Using nonqualified tape drives" on page 325
- "Enabling and disabling persistent reservations" on page 329

Understanding tape devices

About tape devices

When you install a tape drive or tape stacker on a filer according to the tape drive manufacturer's instructions, Data ONTAP creates tape devices associated with the tape drive or stacker. A device is created for each combination of rewind and compression capability. Therefore, a tape drive or tape stacker can have several tape devices associated with it. You must specify a tape device to move, write, or read tapes.

Data ONTAP detects tape drives and tape stackers attached before or after bootup and assigns logical numbers and tape devices to them.

Kinds of tape devices

The following kinds of tape devices are available:

- A local tape device is on the filer that is performing a tape operation.
- A remote tape device is on a NetApp or Solaris machine that
  - Is not the machine that is performing a tape operation, but is connected through the network to a host that is performing the tape operation
  - Is running the `rmt` protocol
  - Has a trust relationship with the filer that is performing the tape operation

Note

You cannot use tape devices associated with tape stackers or tape libraries (medium changers) on a remote Solaris system.

Tape device name format

The format of a tape device name is as follows:

`[`*`remote_host`*`:]`*`rewind_type`* `st` *`drive_number compression_type`*

remote_host is optional. You specify a remote host filer if you want to use a tape drive that is attached to that host. You must follow the remote host name with a colon (:).

rewind_type is the rewind type, as shown in the following table.

Rewind type	Description
`r`	Rewind. Data ONTAP rewinds the tape after it finishes writing the tape file.
`nr`	No rewind. Data ONTAP does not rewind the tape after it finishes writing the tape file. If you want to write multiple tape files on the same tape, specify no-rewind tape device names.
`ur`	Unload/reload. You can use this only if you are performing operations on tapes in tape stackers. When you use these devices, the stacker unloads the tape when it reaches the end of a tape file, and then loads the next tape if there is one. Use this rewind type only under the following circumstances: ◆ The tape drive associated with this device is housed in a tape stacker or a medium changer that is set to stacker mode. ◆ The tape drive associated with this device is attached to a filer. ◆ Sufficient tapes for the operation that you are performing are available in the stacker tape sequence defined for this tape drive.

Note

If you record a tape using a no-rewind device, you must rewind the tape before you read it.

`st` is the standard designation for a tape drive.

drive_number is the drive number. It is incremented for each tape drive; if there is only one tape drive attached to the filer or remote host, the number is 0. The assigned drive number persists through reboots.

compression_type is a drive-specific code for the density of data on the tape and the type of compression. The values for *compression_type* are shown in the following table.

Compression type	Description
a	Highest compression
h	High compression
l	Low compression
m	Medium compression

Examples: `nrst0a` specifies a no-rewind device on tape drive 0 using the highest compression.

`remfiler:nrst0a` specifies a no-rewind device on tape drive 0 on the remote host remfiler that uses the highest compression.

Caution

When using the `urst` device with the `dump` or `restore` command, ensure that the tape drives it is used with are stackers and that there are sufficient tapes in the stacker sequence. Otherwise, the tape drives involved will terminate the command sequence or overwrite the same tape multiple times.

Sample listing of tape devices

The following example shows the tape devices associated with a Digital™ DLT7000:

```
rst0l - rewind device, format is: 81633 bpi 40 GB (w/comp)
nrst0l - no rewind device, format is: 81633 bpi 40 GB (w/comp)
urst0l - unload/reload device, format is: 81633 bpi 40 GB (w/comp)
rst0m - rewind device, format is: 85937 bpi 35 GB
nrst0m - no rewind device, format is: 85937 bpi 35 GB
urst0m - unload/reload device, format is: 85937 bpi 35 GB
rst0h - rewind device, format is: 85937 bpi 50 GB (w/comp)
nrst0h - no rewind device, format is: 85937 bpi 50 GB (w/comp)
urst0h - unload/reload device, format is: 85937 bpi 50 GB (w/comp)
rst0a - rewind device, format is: 85937 bpi 70 GB (w/comp)
nrst0a - no rewind device, format is: 85937 bpi 70 GB (w/comp)
urst0a - unload/reload device, format is: 85937 bpi 70 GB (w/comp)
```

The abbreviations in the example are defined as follows:

- bpi—bits per inch; this is the density of the tape.
- GB—gigabytes; this is the capacity of the tape.
- w/comp—with compression; this shows the tape capacity with compression.

Displaying tape device statistics

Why to display tape drive statistics

It can be helpful to display tape device statistics to understand tape performance and check usage patterns. You can zero out the statistics reading and restart the process of displaying the statistics whenever you want.

How to display tape drive statistics

To display the statistics for a specified tape device, complete the following step.

Step	Action
1	Enter the following command: **`storage stats tape tape`** *tape* is the name of a tape device.

Example: `storage stats tape nrst01`

Example of the output of the storage stats tape command

The following example shows the statistics for the tape device nrst01.

```
filerA> storage stats tape nrst01

Bytes Read: 71471104
Bytes Written: 382147584
Command        Num issued  Max (ms)  Min (ms)  Avg (ms)
-------        ----------  --------  --------  --------
WRITE - Total         2518      1927         2        24  6269 KB/s
   44-48KB             897       372         2         6  6531 KB/s
   60-64KB             421      1927         3        13  4796 KB/s
   128-132KB           800       131         8        19  6761 KB/s
   508KB+              400       481        32        83  6242 KB/s
READ - Total          1092      1570         5        14  4582 KB/s
   60-64KB              92      1390         5        25  2493 KB/s
   64-68KB            1000      1570         5        13  4958 KB/s
WEOF                     5      2827      2787      2810
FSF                      1     13055     13055     13055
BS                       0         0         0         0
FSR                      2      1390         5       697
BSR                      1        23        23        23
REWIND                   9     67606        94     22260
```

```
ERASE                    0          0          0          0
EOM                      1      63223      63223      63223
```

How to zero out tape drive statistics

To zero out the statistics so that only tape device statistics from the time this command is entered will be displayed the next time you enter the `storage stats tape` command, complete the following step.

Step	Action
1	Enter the following command: storage stats tape zero *tape* *tape* is the name of a tape drive.

Example: `storage stats tape zero nrst0l`

Displaying supported tape drives

Why to display supported tape drives

Before connecting a tape drive to a filer, make sure the tape drive is supported by the filer.

How to display supported tape drives

To display a list of the tape drives supported by the filer, complete the following step.

Step	Action
1	Enter the following command: **`storage show tape supported [-v]`** `-v` is an option that displays more complete information about each tape drive.

Sample output of the storage show tape supported command

The following example shows the output of the `storage show tape supported` command.

```
storage show tape supported
Supported Tapes
------------------------
Exabyte 8500C 8mm
Exabyte 8505 8mm
Digital DLT2000
Quantum DLT2000
```

Sample output of the storage show tape supported -v command

The following example shows the output of the `storage show tape supported` command with the `-v` option.

```
storage show tape supported -v
Supported Tapes
------------------------
Exabyte 8500C 8mm
Density Compression
```

```
Setting   Setting
------- -----------
 0x14      0x00      EXB-8200  2.5GB
 0x90      0x10      EXB-8200C (w/compression)
 0x15      0x00      EXB-8500  5.0GB
 0x8c      0x10      EXB-8500C (w/compression)

Exabyte 8505 8mm
Density Compression
Setting   Setting
------- -----------
 0x14      0x00      EXB-8200  2.5GB
 0x90      0x10      EXB-8200C (w/compression)
 0x15      0x00      EXB-8500  5.0GB
 0x8c      0x10      EXB-8500C (w/compression)

Digital DLT2000
Density Compression
Setting   Setting
------- -----------
 0x00      0x00      default density
 0x17      0x00      42500 bpi 2.6GB
 0x18      0x00      42500 bpi 6.0GB
 0x19      0x10      62500 bpi 20 GB (w/comp)

Quantum DLT2000
Density Compression
Setting   Setting
------- -----------
 0x00      0x00      default density
 0x17      0x00      42500 bpi 2.6GB
 0x18      0x00      42500 bpi 6.0GB
 0x19      0x10      62500 bpi 20 GB (w/comp)
```

Assigning tape aliases

Why to assign tape aliases

When you use more than one tape drive (or tape stacker or medium changer) on a filer, Data ONTAP assigns the addresses on a first-come, first-served basis when the software is installed. However, if a filer shares tape drives, or if tape drives are moved around, the software-assigned addresses do not move with the physical devices. After reboot, these addresses are not necessarily reassigned in the same order.

If you use large or multi-filer configurations, or if your filers share backup tape devices, the filers may assign different names to the same tape drives. Tape aliasing enables you to make sure that the same name is always associated with the same tape drive.

What tape aliasing does

Tape aliasing provides a correspondence between the logical names of backup devices (for example, `st0` or `mc1`) and a name permanently assigned to a port or to a tape drive or medium changer.

Note

`st0` and `st00` are considered different logical names.

You can use tape aliases as arguments to the `dump`, `restore`, and `snapmirror store` commands.

Note

Logical and worldwide names are used only to access a device. Once accessed, the device returns all error messages using the physical path name.

Names available for aliasing

The two kinds of names available for aliasing are physical path names (PPNs) and worldwide names (WWNs).

What physical path names are

Physical path names (PPNs) are the numerical address sequences that Data ONTAP assigns to tape drives and tape libraries based on which SCSI-2/3 adapter or switch (specific location) they are connected to on the filer. PPNs are also known as electrical names.

Format of the PPN

Direct-attached devices: PPNs of direct-attached devices use the following format:

host_adapter.device_id_lun

For example, the PPN 8.6 indicates that the host adapter number is 8, the device ID is 6, and the logical unit number (LUN) is 0.

Note

On disk drives, the LUN is always 0 and is therefore never shown. On tape drives and medium changers, the LUN is also not shown if it is 0.

Switch-attached devices: PPNs of switch-attached devices use the following format:

switch:port_id.device_id_lun

For example, the PPN MY_SWITCH:5.3L2 indicates that the tape drive that is connected to port 5 of a switch called MY_SWITCH is set with SCSI ID 3 and has the LUN 2.

The LUN is determined by the drive itself. Both Fibre Channel and SCSI tape drives and libraries, as well as disks, have PPNs.

Examples using PPNs

In the following example, the `dump` command is using the tape device name of a tape drive:

```
dump 0f /dev/nrst0a /vol/vol0
```

In the following example, the same `dump` command is using the PPN of the tape drive:

```
dump 0f /dev/nr.MY_SWITCH:5.6.a /vol/vol0
```

PPNs of tape drives and libraries do not change unless the name of the switch changes, the tape drive or library moves, or the tape drive or library is reconfigured. PPNs remain unchanged after reboot.

For example, if a tape drive named MY_SWITCH:5.3L2 is removed and a new tape drive with the same SCSI ID and LUN is connected to port 5 of switch MY_SWITCH, the new tape drive would be accessible via the name MY_SWITCH:5.3L2.

What worldwide names are

Tape drives and libraries are assigned worldwide names (WWNs) at the time of manufacture. WWNs are similar to the media access control (MAC) addresses on Ethernet cards. All Fibre Channel devices have WWNs, but SCSI-attached devices do not have WWNs and cannot be accessed by them.

Accessing a tape drive or library via the WWN allows multiple filers to track the same device. Depending on whether a tape drive is connected to a Fibre Channel switch or hub or is directly attached to a Fibre Channel adapter, different filers may have different PPNs for the same device. Using the WWN in these cases eliminates any confusion.

Also, if a switch is renamed or a tape drive is moved on the filer, the WWN of the tape drive or library does not change, and scripts or backup programs do not need to change the name of the tape drive or library to which they are backing up.

Format of the WWN

The WWN of a tape device uses the following format:

WWN[#:###:######:######]L##

is a hexadecimal character and L is the LUN of the device. If the LUN is 0, it is not shown.

Each WWN consists of eight bytes, and the format for the WWN is not case-sensitive.

Examples using WWNs

In the following example, the `dump` command uses the logical name of a tape drive:

```
dump 0f /dev/nrst0a /vol/vol0
```

In the following example, the same `dump` command uses the worldwide name of the tape drive:

```
dump 0f /dev/nr.WWN[2:000:00e08b:01523e].a /vol/vol0
```

Displaying existing aliases

To display existing aliases, complete the following step:

Step	Action

1	Enter the following command:
	`storage alias`
	Result: A display like the following appears:
	`Alias Mapping`
	`----------`
	`st0 MY_SWITCH:5.3L3`
	`st2 MY_SWITCH:5.4L6`
	`mc1 2:4e3:38fe3f:758eab`
	`mc348 MY_SWITCH:5.3L0`
	In this example, the display shows that there are two tape drives and two medium changers attached to the filer. Tape drives st0 and st2 and medium changer mc348 are attached to port 5 of the switch MY_SWITCH. Medium changer mc1 has the worldwide name 2:4e3:38fe3f:758eab.

Displaying information about tape drives or libraries

To display information about tape drives and tape libraries (medium changers) to help you assign tape aliases, complete the following step.

Step	Action
1	Enter the following command: **`storage show {tape | me} [{`*`alias`*` | `*`PPN`*` | `*`WWN`*`}]`** *alias* is the logical name of the tape drive or medium changer. *PPN* is the physical path name. *WWN* is the worldwide name.

Examples:

```
storage show tape

     Tape Drive:        MY_SWITCH:5.3L4
     Description:       Quantum DLT7000
     Serial Number:     12345679
     World Wide Name:   WWN[2:333:444444:555555]L4
```

```
     Alias Name(s):     st0  st1  st2  st3

     Tape Drive:        MY_SWITCH:5.3L5
     Description:       Quantum DLT7000
     Serial Number:     12345678
     World Wide Name:   WWN[2:777:888888:999999]L5
     Alias Name(s):     st10  st11  st12  st13

storage show tape st0
     Tape Drive:        MY_SWITCH:5.3L4
     Description:       Quantum DLT7000
     Serial Number:     12345679
     World Wide Name:   WWN[2:333:444444:555555]L4
     Alias Name(s):     st0  st1  st2  st3

storage show tape MY_SWITCH:5.3L4
     Tape Drive:        MY_SWITCH:5.3L4
     Description:       Quantum DLT7000
     Serial Number:     12345679
     World Wide Name:   WWN[2:333:444444:555555]L4
     Alias Name(s):     st0  st1  st2  st3
```

Assigning tape aliases

To assign an alias to a tape drive or medium changer, complete the following step.

Step	Action
1	Enter the following command: **`storage alias [`*`alias`* `{`*`PPN`*` | `*`WWN`*`}]`** *alias* is the logical name of the tape drive or medium changer to which you want to add the alias. *PPN* is the physical path name to which you want to assign the tape drive or medium changer. *WWN* is the worldwide name to which you want to assign the tape drive or medium changer.

Example:
```
storage alias st0 MY_SWITCH:5.3L3
     storage alias mc80 WWN[2:4e3:38fe3f:758eab]
```

Result: The tape device st0 is assigned the PPN MY_SWITCH:5.3L3. The medium changer mc80 is assigned the WWN WWN[2:4e3:38fe3f:758eab].

Removing tape aliases

To remove an alias from a tape drive or medium changer, complete the following step.

Step	Action
1	Enter the following command: **`storage unalias {alias | -a | -m | -t}`** *alias* is the logical name of the tape drive or medium changer from which you want to remove the alias. `-a` removes all aliases. `-m` removes the aliases from all medium changers. `-t` removes the aliases from all tape drives.

Example:
```
storage unalias st0
storage unalias mc80
```

Result: The aliases are removed from the tape device st0 and from the medium changer mc80.

Propagating tape aliases to multiple filers

If you need to use the same set of tape drives to back up more than one filer, you can save the tape alias information in a file.

To propagate tape aliases to multiple filers, complete the following steps.

Step	Action

1	Create a file named *tape_alias* containing the tape alias information.
	Example:
	`storage unalias -a`
	`storage alias st0 8.6`
	`storage alias st1 8.7`
	`storage alias mc0 8.1`

2	Copy the file to the root volume of each filer.

Step	Action
3	Execute the following command on each filer: **source /vol/*root_volume_name*/*tape_alias*** *root_volume_name* specifies the root volume. **Result:** The filers all contain the same configuration information.

To ensure that multiple filers assign the same alias to a tape drive or medium changer, you can type the same set of storage alias commands on each filer.

Alternatively, UNIX users can use a shell script like the following to propagate the source script information to the filers.

```
#!/bin/sh
# Check for the source file
if [ "$#" != "1" ]
then
     echo "Usage: $0 <source_file>"
     exit 1
fi

if [ ! -r $1 ]
then
     echo "Cannot open source file \"$1\""
     exit 1
fi

while [ 1 ]
do

     echo Hit ctrl-c to terminate program when all filers have been
entered.

# Gather up filer and passwd from user
     printf "File Server: "
     read FILER
     printf "Password: "
     read PASSWD
     stty -echo
     stty echo
     printf "\n"

     # Now issue the commands in the source script to
     # the remote filer.

     while read cmd
     do
```

```
        echo Issuing command \"$cmd\" to filer $FILER
        rsh $FILER -l root:$PASSWD "$cmd" < /dev/null
    done < $1
    printf "\n"
done
```

Adding Fibre-Channel-attached tape drives and libraries

You can add Fibre-Channel-attached tape drives and libraries to filers dynamically (without taking the filers offline). When the new medium changer is added, the filer detects its presence and adds it to the configuration. If the medium changer is already referenced in the alias information, no new logical names are created. If the library is not referenced, the filer creates a new alias for the medium changer.

Displaying tape device and tape drive information

About displaying tape device information

Before you back up to a tape device attached to the filer, you might want to verify that the filer detects the tape drive associated with the device and verify the available tape device names associated with the tape drive.

Qualified versus nonqualified tape drives

Qualified tape drives are tape drives that Network Appliance has tested and found to work properly with filers. A list of qualified tape drives is available on the NOW site at

http://now.netapp.com/NOW/knowledge/docs/olio/cert_matrices/

Nonqualified tape drives are not supported by Network Appliance. You can use a nonqualified tape drive by qualifying the tape drive dynamically, as described in "Qualifying tape drives dynamically" on page 321, or by making it emulate a qualified tape drive, as described in "Using nonqualified tape drives" on page 325.

Information about qualified tape drives in tape libraries

Qualified tape drives that are in tape stackers or libraries are listed by drive. For example, a Breece-Hill library is not listed in all displays, whereas the DLT7000 tape drives that it contains are listed.

Types of information available

You can display information about

- Qualified tape drives
- Nonqualified tape drives
- Tape stackers
- Tape drive connections to the filer

Displaying information about tape drives

To display information about all tape drives on a filer, complete the following step.

Step	Action

1	Enter the following command:
	`sysconfig -t`
	Result: A display like the following appears. The numbers following "Tape drive" show the slot on the filer that the drive is attached to, followed by the drive's SCSI ID. In the following example, the Exabyte 8900 has SCSI ID 1 and is attached to a controller in slot 0b. Information about the Digital DLT7000 includes bpi specifications provided by the manufacturer.
	Note
	Compression capacity in the display is estimated; actual capacity depends on how much the data being written to the tape can be compressed.
	`Tape drive (0b.1)  Exabyte 8900 8mm`
	`rst0l  -  rewind device,        format is: EXB-8500  5.0GB(readonly)`
	`nrst0l -  no rewind device,     format is: EXB-8500  5.0GB(readonly)`
	`urst0l -  unload/reload device, format is: EXB-8500  5.0GB(readonly)`
	`rst0m  -  rewind device,        format is: EXB-8500C (w/compression)`
	`nrst0m -  no rewind device,     format is: EXB-8500C (w/compression)`
	`urst0m -  unload/reload device, format is: EXB-8500C (w/compression)`
	`rst0h  -  rewind device,        format is: EXB-8900  10.0GB`
	`nrst0h -  no rewind device,     format is: EXB-8900  10.0GB`
	`urst0h -  unload/reload device, format is: EXB-8900  10.0GB`
	`rst0a  -  rewind device,        format is: EXB-8900C (w/compression)`
	`nrst0a -  no rewind device,     format is: EXB-8900C (w/compression)`
	`urst0a -  unload/reload device, format is: EXB-8900C (w/compression)`
	`Tape drive (5.3)  Digital DLT7000`
	`rst1l  -  rewind device,        format is: 81633 bpi 40 GB (w/comp)`
	`nrst1l -  no rewind device,     format is: 81633 bpi 40 GB (w/comp)`
	`urst1l -  unload/reload device, format is: 81633 bpi 40 GB (w/comp)`
	`rst1m  -  rewind device,        format is: 85937 bpi 35 GB`
	`nrst1m -  no rewind device,     format is: 85937 bpi 35 GB`
	`urst1m -  unload/reload device, format is: 85937 bpi 35 GB`
	`rst1h  -  rewind device,        format is: 85937 bpi 50 GB (w/comp)`
	`nrst1h -  no rewind device,     format is: 85937 bpi 50 GB (w/comp)`
	`urst1h -  unload/reload device, format is: 85937 bpi 50 GB (w/comp)`
	`rst1a  -  rewind device,        format is: 85937 bpi 70 GB (w/comp)`
	`nrst1a -  no rewind device,     format is: 85937 bpi 70 GB (w/comp)`
	`urst1a -  unload/reload device, format is: 85937 bpi 70 GB (w/comp)`

Displaying information about tape medium changers

To display information about tape medium changers, complete the following step.

Step	Action

1	Enter the following command:
	sysconfig -m
	Result: A display like the following appears. In this example, a tape stacker with SCSI ID 4 is attached to slot 6 of the filer.
	`Medium changer (6.4)  BHTi     Quad 7`
	`mc0 - medium changer device`
	Note
	If autoload on the medium changer is set to On, the medium changer information might not appear.
	Note
	The filer supports only tape medium changers that are compliant with SCSI-2 specifications.

Displaying information about tape drive connections to the filer

To display information about tape drive connections to the filer, complete the following steps.

Step	Action
1	Enter the following command: **sysconfig -v**
2	Look for the information about the SCSI adapter to which the tape devices are attached.

Example: This example shows a tape medium changer with SCSI ID 6 and a tape drive with SCSI ID 4 attached to slot 6 of the filer. The SCSI firmware is 2.26, and the SCSI adapter clock rate is 60 MHz. For each drive, the following information appears:

- SCSI ID

- Vendor ID
- Product ID
- Firmware version

```
slot 6: SCSI Host Adapter 6 (QLogic ISP 1040B)
   Firmware Version 2.26      Clock Rate 60MHz.
   6: BHTi    Quad 7          1.41
   4: QUANTUM DLT7000         1B41
```

Understanding tape files

What a tape file is

All data on a tape is stored in tape files. Tape files are separated by file marks on the tape.

About tape file names

Tape files have no names.

About specifying a tape file

You specify a tape file by its position on the tape.

About reading and writing tape files

You write a tape file using a tape device that you specify. When you read the tape file, you must specify a device that has the same compression type that you used to write it. The compression type is indicated by the last character in the tape device name, for example, the `a` in `nrst0a`.

For example, if you have two tape drives of the same model with tape drive numbers 1 and 2 and you write a file on `nrst0a`, you can read it safely on the other tape drive using `rst1a`, because devices of the same model have the same compression type (type `a`).

Controlling tape drives

About controlling tape drives

You can control tape drives with the `mt` command to move and position the tape in several ways. This section describes how to perform the following tasks:

- Move a tape to the end of data. You do this to append a backup.
- Skip forward over files. You do this to access a particular tape file.
- Skip backward over files. You do this to access a particular tape file.
- Append a backup. You do this to save tape if you have small backups.
- Rewind a tape. You do this to get to the beginning of the tape after using a no-rewind device.
- Take a tape drive offline. You do this to service a tape drive.
- Display status information. You do this to find out whether you can read the tape with a device or to verify that a tape drive is not in use.

The mt command syntax

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

`mt {-f|-t} device command [ count ]`

The `-f` and `-t` options are interchangeable; they indicate that the next argument is a device.

device is a tape device.

command is an `mt` command. The following table discusses the most common `mt` commands.

count specifies the number of times to execute a command that supports multiple operations.

This section discusses the `mt` commands listed in the following table. For a description of all the `mt` commands, see the mt(1) man page.

Command	Task
`eom`	Position the tape to the end of the data or the end of the medium if the tape is full.
`fsf`	Move the tape forward, skipping a specified number of files.

Command	Task
`bsf`	Move the tape backward, skipping a specified number of files.
`rewind`	Rewind the tape.
`offline`	Rewind the tape and unload the tape medium if possible.
`status`	Display information about a device and the drive associated with it.

Note

Use `nrst` devices for all tape status and movement operations. Using other rewind types might produce unwanted side effects.

Caution

When you use the `urst` device with the `mt` command, the tape drives it is used with must be stackers and there must be sufficient tapes in the stacker sequence. Otherwise, the tape drives involved will terminate the command sequence or overwrite the same tape multiple times.

Moving a tape to the end of data

You move a tape to the end of data if you want to append data on a tape.

To move a tape to the end of data, or end of medium if the tape is full, complete the following step.

Step	Action
1	Enter the following command: **`mt -f device eom`** *device* is the name of a no-rewind tape device.

Example: `mt -f nrst0a eom`

Note

If you use a rewind or unload/reload tape device, this command rewinds the device, moves the tape to the beginning of data, and unloads it if possible.

Moving forward to a file

You move forward to access a particular tape file further along the tape. You can skip over a specified number of file marks and stop at the end-of-tape side of a file mark. This puts the tape drive head at the beginning of a file.

To move forward to the beginning of a tape file, complete the following step.

Step	Action
1	Enter the following command: **mt -f *device* fsf *n*** *device* is the name of a tape device used on the tape. *n* is the number of tape file marks you want to skip over going forward. The tape moves to the beginning of the *n*th file after the file where it is when the command is issued.

Example: If you enter the following command in the middle of the third file on the tape, it moves the tape to the beginning of the eighth file on the tape:

```
mt -f nrst0a fsf 5
```

Moving backward to the beginning of a file

You move backward to access a particular tape file further back on the tape.

To move backward to the beginning of a tape file, complete the following steps.

Step	Action
1	Enter the following command: **mt -f *device* bsf *n*** *device* is the name of a tape device used on the tape. *n* is the number of tape file marks you want to skip over going backward. The tape moves to the beginning of the *n*th file before the file where it is when the command is issued.

Example: If you enter the following command in the middle of the fifth file on the tape, it moves the tape to the beginning of the second file on the tape:

```
mt -f nrst0a bsf 3
```

Rewinding a tape

You rewind a tape to get to the beginning of the tape when using a no-rewind device. You do this before performing a restore of a backup that was done using a no-rewind device, provided that there is no other data before the backed-up data on the tape and it is not unloaded.

To rewind a tape, complete the following step.

Step	Action
1	Enter the following command: **`mt -f device rewind`** *device* is the name of a tape device used on the tape.

Example: `mt -f nrst0a rewind`

Taking a tape drive offline

You take a tape drive offline to service it.

To rewind the tape and take the tape drive offline by unloading the tape, complete the following step.

Step	Action
1	Enter the following command: **`mt -f device offline`** *device* is the name of a tape device used on the tape.

Example: `mt -f nrst0a offline`

Displaying status information

You display status information to find out whether you can read with a device or to verify that a tape drive is not in use.

To display status information about a tape device and the drive associated with it, complete the following step.

Step	Action
1	Enter the following command: **`mt -f device status`** *device* is the name of a tape device.

Example: `mt -f nrst0a status`

Result: A display like the following appears:

```
Tape drive: Quantum DLT7000
Status: ready, write enabled
Format: 85937 bpi 70 GB (w/comp)
fileno = 0  blockno = 95603  resid = 0
```

The fields in the display are as follows:

- Tape drive—The model of the tape drive.
- Status—Whether the tape drive is ready and write-enabled.
- Format—The number of bytes per inch, total capacity in gigabytes, and whether data compression is used.
- fileno—The current tape file number. Numbering starts at 0.
- blockno—The current block number.
- resid—The number of bytes that the drive attempted to write or read but could not because the end of tape was reached.

Qualifying tape drives dynamically

What a qualified tape drive is

A qualified tape drive is a tape drive that has been tested and found to work properly on Network Appliance filers. A qualified tape drive is in the Data ONTAP kernel's internal tape qualification list.

How to qualify a tape drive dynamically

You can qualify a nonqualified tape drive by entering a valid tape qualification file in the filer's /etc/tape_config directory. You get the information you need for this file from the tape vendor's drive documentation. You can use a text editor to create the tape qualification file. There can be multiple files in the directory, one for each model being qualified.

How to format the tape qualification file

The /etc/tape_config directory contains a sample tape qualification file, which includes an explanation of the requirements for a tape qualification file, a list of the default SCSI command timeout values used by the tape drive, and an example of a tape qualification file.

When you create a new tape qualification file, you can use any valid file name, except a file name that begins with a period (.).

The format of the tape qualification file is as follows:

vendor_id (string, limit of 8 bytes)—The vendor ID as reported by the `SCSI Inquiry` command.

product_id (string, limit of 16 bytes)—The product ID as reported by the `SCSI Inquiry` command.

id_match_size (number)—The number of bytes of the product ID to be used for matching to detect the tape drive to be identified, beginning with the first character of the product ID in the Inquiry data.

vendor_pretty (string, limit 16 bytes)—If this parameter is present, it is specified by the string displayed by `sysconfig -v` or `sysconfig -t`; otherwise, the INQ_VENDOR_ID is displayed.

product_pretty (string, limit 16 bytes)—If this parameter is present, it is specified by the string displayed by `sysconfig -v` or `sysconfig -t`; otherwise, the INQ_VENDOR_ID is displayed.

Note

The *vendor_pretty* and *product_pretty* fields are optional but should be used together. If you specify a value for one of these fields, you must also specify a value for the other.

You must specify a description, density code, and compression algorithm for *l*, *m*, *h*, and *a*, as follows:

{*l* | *m* | *h* | *a*}*_description*= (string, limit 16 bytes)—The string to print for `sysconfig -t` that describes characteristics of the particular density setting.

{*l* | *m* | *h* | *a*}*_density*= (hex codes)—The density code to be set in the SCSI mode page block descriptor corresponding to the desired density code for *l*, *m*, *h*, or *a*.

{*l* | *m* | *h* | *a*}*_algorithm*= (hex codes)—The compression algorithm to be set in the SCSI Compression Mode Page corresponding to the density code and the desired density characteristic.

The following fields are optional:

autoload= (Boolean yes/no)—Set this field to *yes* if the tape drive has an automatic loading feature; that is, after you insert a tape cartridge, the tape drive becomes ready without the need to execute a SCSI load (start/stop unit) command. The default for this field is *no*.

*cmd_timeout_0x**—Individual timeout value. This field must only be used if you want to specify a different timeout value from the one being used as a default by the tape driver. The sample file lists the default SCSI command timeout values used by the tape driver. The timeout value can be expressed in minutes (m), seconds (s), or milliseconds (ms).

Example of a tape configuration file

The following example shows a tape configuration file for the tape drive Tape Drive 2001.

```
vendor_id="TAPECO"
product_id="TAPEDRIVE2001"
id_match_size=11

vendor_pretty="Tape Company"
product_pretty="Tape Drive 2001"

  l_description="2 GB"
  l_density=0x01
  l_algorithm=0x81
```

```
  m_description="4 GB (compression)"
  m_density=0x01
  m_algorithm=0x82

  h_description="6 GB"
  h_density=0x10
  h_algorithm=0x88

  a_description="12 GB (compression)"
  a_density=0x10
  a_algorithm=0x89

autoload="yes"
# no entry for autoload is the same as autoload="no"

cmd_timeout_0x01=5m
cmd_timeout_0x12=700ms
cmd_timeout_0x1b=60s
```

How the filer qualifies a new tape drive dynamically

The filer's /etc/tape_config directory is created automatically when the filer boots. When you add a tape qualification file to this directory, the filer checks the file's format at the next boot time or the next time any tape is accessed. If the format is valid, the information is entered into the internal tape qualification table and a success message similar to the following is printed to the console and system log. Information about the tape persists as long as the file is in the directory or until the file is altered.

```
Dynamic Tape Qualification file:/etc/tape_config/filename format
check was successful. Internal tape qualification table was updated
with Tape Drive product information.
```

If the format is incorrect, an error message similar to one of the following is printed to the console and system log.

```
Dynamic Tape Qualification file: /etc/tape_config/filename has
missing or badly formatted required key(s).
Dynamic Tape Qualification file: /etc/tape_config/filename has a
format error in the information appended to the required key.
```

When you connect a tape drive to the filer, the filer looks for a *product_id* match between information obtained during the tape discovery process and information contained in the internal tape qualification table. If a match is discovered, the drive is marked as qualified and the filer can access the tape drive. If a match is not found, the tape drive remains in the unqualified state and is not accessed.

Qualified SCSI drives must be attached to the filer before reboot to be recognized; qualified Fibre Channel drives are recognized whenever they are attached.

Using nonqualified tape drives

How you can use a nonqualified tape drive

You can use a nonqualified tape drive (one that is not on the list of qualified tape drives) on a filer if it can emulate a qualified tape drive. It is then treated as though it were a qualified tape drive.

Displaying information about nonqualified tape drives

To display information about nonqualified tape drives, complete the following steps.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="3">1</td><td>If the filer...</td><td>Then...</td></tr>
<tr><td>Has accessed the tape drive through the <code>dump</code> or <code>mt</code> command</td><td>Go to Step 3.</td></tr>
<tr><td>Has not accessed the tape drive through the <code>dump</code> or <code>mt</code> command</td><td>Go to Step 2, then proceed to Step 3.</td></tr>
<tr><td>2</td><td colspan="2">Enter the following command to access the tape drive:
<code>mt -f device status</code>
device is any device that contains the tape drive number that you think is assigned to the tape drive.
Example: <code>mt -f nrst1a status</code></td></tr>
<tr><td>3</td><td colspan="2">Enter the following command:
<code>sysconfig -t</code>
Result: If the filer has registered a tape drive as emulating a qualified tape drive, it displays a message like the following:
<code>Tape drive (6.5) DLT9000 emulates Digital DLT7000.</code>
If the filer has not registered a tape drive as emulating a qualified tape drive, it displays a message like the following:
<code>Tape drive (6.5) DLTXXXX (Non-qualified tape drive)</code></td></tr>
</table>

Getting tape drive information for emulation

You must supply the following information about the nonqualified tape drive from your tape drive documentation to emulate a qualified tape drive:

- What qualified tape drive the nonqualified tape drive can emulate.
- The vendor ID string. This is a SCSI string and should be in the SCSI section of your tape drive manual.
- The product ID string. This is a SCSI string and should be in the SCSI section of your tape drive manual.

Note

Your nonqualified tape drive must be able to emulate a qualified tape drive for the emulation to succeed.

Emulating a qualified tape drive

To emulate a qualified tape drive, complete the following steps.

Step	Action
1	Make sure that you have a tape adapter available on the filer.
2	Halt the filer.
3	Connect the tape drive to the filer according the tape drive manufacturer's instructions.
4	Turn on the tape drive.
5	Turn on the filer. **Result:** When the filer boots, an error message tells you that the tape drive is unsupported. It also contains the vendor ID and the product ID for the tape drive.
6	Record the vendor ID and product ID from Step 5 for later use.
7	Enter the following command to determine the highest tape drive number on your filer: `sysconfig -t`

Step	Action
8	Enter the following command to create the filer /etc/cloned_tapes file: **`mt -f nrst`*`n`*`a status`** *n* is one plus the highest tape drive number from Step 7. **Example:** `mt -f nrst2a status`
9	Open the filer's /etc/cloned_tapes file in a text editor on a client that can access it.
10	For each nonqualified tape drive, create a line with either of following formats: *clone_vendor_ID clone_product_ID* `EMULATES` *vendor_ID product_ID* *clone_product_ID* `EMULATES` *product_ID* *clone_vendor_ID* is the vendor of the nonqualified tape drive. *clone_product_ID* is the model number of the nonqualified tape drive. *vendor_ID* is the vendor of a qualified tape drive that you want the nonqualified tape drive to emulate. *product_ID* is the model number of a qualified tape drive that you want the nonqualified tape drive to emulate. **Example:** The following entry in the /etc/cloned_tapes file enables the filer to treat the nonqualified Quantum DLT9000 tape drive as a clone of the qualified Quantum DLT7000 tape drive: `QUANTUM DLT9000 EMULATES QUANTUM DLT7000`
11	Enter the following command to put the emulation into effect: **`mt -f rst`*`n`*`a status`** *n* is one plus the highest tape drive number from Step 7. **Example:** `mt -f rst2a status`

Step	Action
12	Verify that the drive appears, along with all your previously installed devices, by entering the following command: **sysconfig -t**

Enabling and disabling persistent reservations

What tape reservations are

Because multiple filers can share access to one tape drive or medium changer, it is advantageous to prevent more than one filer from accessing a tape drive or a medium changer at a time. Tape reservation solves this problem by enabling SCSI-persistent reservations for all Fibre-Channel-attached tape drives and medium changers. These reservations provide a method for tape drives and medium changers to accept commands from only one filer at a time.

These reservations cannot be enabled or disabled on an individual device basis.

Why tape reservations are not automatically enabled

Because not all tape drives and medium changers implement SCSI-persistent reservations correctly, this option is disabled by default to prevent problems from occurring with these devices.

How to enable persistent reservations

To turn on persistent reservations, complete the following step.

Step	Action
1	Enter the following command: `options tape.persistent_reservations on`

When to disable persistent reservations

Enabling this option may cause problems if tape drives or medium changers do not work properly. If tape commands report the device is reserved when no other filers are using the device, this option should be disabled.

How to disable persistent reservations

To turn off persistent reservations, complete the following step.

Step	Action
1	Enter the following command: `options tape.persistent_reservations off`

Data Backup to Tape 9

About this chapter

This chapter describes how to use the `dump` command to back up data on the filer to tape.

Topics in this chapter

This chapter discusses the following topics:

- "Introduction to data backup to tape" on page 332
- "How the dump command works" on page 336
- "Preparing to use the dump command" on page 338
- "Using the dump command" on page 342
- "Appending backups to a tape" on page 365
- "Examples of the dump command" on page 366
- "Verifying the contents of a dump command backup" on page 379
- "Checking the status of a dump command backup" on page 380
- "Restarting a dump command backup" on page 382
- "Deleting restartable dump command backups" on page 384

Introduction to data backup to tape

About data backup

Although data backup can also mean copying data to standard output, for this chapter, data backup means copying data from disk to tape.

Reasons to back up data from disk to tape

You back up data from disk to tape for the following reasons:

- You can store the backup tapes at an off-site archive to protect the data against natural disasters.
- You can restore data from tape if an application or a user inadvertently corrupts or deletes files that cannot be recovered from snapshots.
- You can restore data from tape after you reinstall the file system on the filer (for example, when migrating to larger disks or converting a single-volume filer to a multivolume filer).

For detailed information

The following sections present information about the types of backups you can perform, how to perform them, and what gets backed up:

- "Types and methods of backup" on page 333
- "What the dump command backs up" on page 335

Types and methods of backup

Types of backups

You can do both full and incremental backups:

- In a full backup, everything is backed up.
- In an incremental backup, only files changed since a previous backup are written to the backup media.

Backup methods

Data ONTAP supports three backup methods:

- Using the Data ONTAP `dump` command.
- Using an NDMP-compliant backup application. For more information about NDMP, see Chapter 11, "Using NDMP Services," on page 425.
- Using commands or a non-NDMP-compliant application from an NFS client or a CIFS client, such as the `tar` or `cpio` command on an NFS client.

Caution

If you use the `tar` or `cpio` command on an NFS client to back up the filer, be aware that some versions of these commands fail with file systems that contain long path names, unusual file names, or hard links.

Comparison of backup methods

The following table compares the backup methods.

	Filer dump command	**NDMP applications**	**Commands or non-NDMP applications on NFS or CIFS client**
Where backup is initiated	Initiates the backup session from the filer console or from a trusted host through the `rsh` command.	Initiate the backup session from the computer running the NDMP application.	Initiate the backup session from an NFS client or a CIFS client.

	Filer dump command	NDMP applications	Commands or non-NDMP applications on NFS or CIFS client
What is backed up	Backs up any files or directories.	Back up any files or directories.	Back up the files or directories that are mounted or shared by the client.
Where data is backed up	Backs up data to tape drives attached to the filer being backed up, to another filer, or to another computer that supports the `rmt` protocol.	Back up data to tape drives attached to the filer being backed up, to the computer running the NDMP application, or to another filer, as well as to NDMP-compliant tape libraries.	Back up data to tape drives attached to the client or, if the client supports the `rmt` protocol, to tape drives attached to the filer. The details are application-dependent.
What attributes are preserved	Preserves both NFS and CIFS attributes.	Preserve both NFS and CIFS attributes.	Preserved attributes are application-dependent.

What the dump command backs up

What the dump command can back up

The `dump` command backs up data so that it can look the same when restored as it did when it was backed up.

The `dump` command can back up the following objects:

- A file
- A directory
- A qtree
- An entire volume

In addition to backing up data in files, the `dump` command can back up the following information about each file, as applicable:

- UNIX GID (Group ID), owner UID (User ID), and file permissions
- UNIX access, creation, and modification time
- File type
- File size
- DOS name, DOS attributes, and creation time
- Windows NT Access Control Lists (ACLs)
- Qtree information

Dump command data format

The data is written to tape in a format compatible with the standard BSD-dump format. This means that you can restore Network Appliance dump tapes using the Solaris `ufsrestore` command. The only data that you cannot restore with this command is the CIFS (Windows NT) attributes, because Solaris does not support them. However, the data itself is accessible.

Dump and restore permissions limitations

If you dump a file that has only Windows NT permissions and restore it into a UNIX-style qtree or volume, the file gets the default UNIX permissions for that qtree or volume.

If you dump a file that has only UNIX permissions and restore it into an NTFS-style qtree or volume, the file gets the default Windows permissions for that qtree or volume. Other dumps and restores preserve permissions.

How the dump command works

What the dump command does

The `dump` command writes file system data from disk to tape. It is optimized for data restoration to a filer using the Data ONTAP `restore` command.

Kinds of backups

There are three kinds of backups:

- Full volume backup, where you back up an entire volume
- Full qtree backup, where you back up an entire qtree
- Miscellaneous backup, where you back up a subtree that is neither an entire volume nor an entire qtree

The backup process

Data ONTAP uses the process described in the following table to back up the object indicated by the dump path.

Stage	Action
1	For less than full volume or full qtree backups, Data ONTAP traverses directories to identify files to be backed up. If you are backing up an entire volume or qtree, Data ONTAP combines this step with Step 2.
2	For a full volume or full qtree backup, Data ONTAP identifies the directories in the volumes or qtrees to be backed up.
3	Data ONTAP writes the directories to tape.
4	Data ONTAP writes the files to tape.
5	Data ONTAP writes the access control list (ACL) information (if applicable) to tape.

How the dump command uses snapshots in backing up data

The `dump` command uses a snapshot of your data for the backup, so you do not need to take the filer or volume offline before initiating the backup.

The `dump` command names each snapshot it creates snapshot_for_backup.*n*.

- *n* is an integer starting at 0.

- Each time the `dump` command creates a snapshot, it increments the integer by 1.
- The filer resets the integer to 0 when it is rebooted.

When Data ONTAP executes multiple `dump` commands simultaneously, the `dump` commands create multiple snapshots. For example, if Data ONTAP is running two `dump` commands simultaneously, you find these snapshots in the volumes from which data is being backed up: snapshot_for_backup.0 and snapshot_for_backup.1.

Note

When you are backing up from a snapshot, `dump` does not create an additional snapshot.

Preparing to use the dump command

About dump command preparations

Before backing up data with the `dump` command, you need to make preparations. You must have sufficient tapes and know how you will enter the `dump` command.

For detailed information

The following topics discuss what you need to do to prepare for using the dump command:

- "Determining the amount of backup data" on page 339
- "Estimating the number of tapes for the backup" on page 340
- "Deciding how to enter the dump command" on page 341

Determining the amount of backup data

About determining the amount of backup data

Before you enter the `dump` command, it is helpful to estimate the amount of backup data so that you can determine the number of tape files and the number of tapes required for the backup.

Note

This method of estimation works only for full backups of volumes. It is impossible to determine how much data is being backed up in an incremental backup.

Estimating the amount of backup data in a volume

To estimate the amount of data in a full backup, complete the following steps.

Step	Action
1	For each item that you want to back up, enter the following command: **`df path_name`** *path_name* is the name of the item.
2	For multiple items, such as multiple volumes, add up the data for each item to determine the total amount of data to be backed up.

Estimating the number of tapes for the backup

Reasons to determine the number of tapes

You should estimate the number of tapes required for the backup before entering the `dump` command for these reasons:

- You can ensure that the `dump` command does not fail because it runs out of tapes.
- You can load the required number of tapes in the tape drives or stackers in advance for an unattended backup. If you do not load enough tapes before entering the `dump` command and you start the `dump` command from the console, Data ONTAP prompts you to load additional tapes. If you start the `dump` command through a Remote Shell connection, you do not see the prompts from Data ONTAP and the `dump` command terminates if there are not enough tapes.

Estimating the number of tapes

To estimate the number of tapes required for the backup, complete the following steps.

Step	Action
1	Determine the capacity of the tape device you are using for the backup by entering the following command: **`sysconfig -t`**
2	Determine the amount of data that needs to be backed up, as described in "Determining the amount of backup data" on page 339.
3	Divide the amount of data by the capacity of a tape.
4	If your estimate indicates that your data will nearly fill the last tape, add a tape to the estimate in case the backup exceeds your estimate. This is especially important when using compression, because compression rates vary based on the data.

Deciding how to enter the dump command

About entering the dump command

You can enter the `dump` command through a Remote Shell connection, such as through the `rsh` command, through a Telnet session accessing the filer console, or through the filer console directly.

Other than potential problems associated with any remote connection, console access through a Telnet session and direct console connection to the filer behave the same way.

Benefits of entering the dump command through a Remote Shell connection

Entering the `dump` command through a Remote Shell connection gives you these benefits:

- When the `dump` command is in progress, you can still use the console to manage the filer. If the `dump` command entered on the console is backing up a large number of files, you cannot use the console for a long time.
- You can start multiple `dump` commands through the `rsh` command.
- Data ONTAP is less likely to inadvertently terminate the `dump` command, especially if it is run in the background from a Solaris system. If you enter a `dump` command on the filer console, it could be terminated by Ctrl-C entered on a host connected to the filer using a Telnet session.
- You can automate filer backups through shell scripts and crontab entries.

Benefits of entering the dump command at the console

If you enter the `dump` command at the console, you can read and respond to screen messages and prompts displayed by the command. For example, the command might prompt you for another tape to complete the backup, whereas a `dump` command entered through a Remote Shell connection does not generate any messages when the command needs user intervention, and terminates instead.

Using the dump command

About the dump command

You can enter the `dump` command any time the tape devices you want to use are free to back up data in a specified path. After the `dump` command is finished, the data in the path is written to tape.

You can run up to eight `dump` commands (depending on the hardware you are using) in parallel on up to eight tape drives, one command per drive. Parallel backups increase throughput.

For detailed information

The dump command is complex and has many options. The following sections discuss how to construct a dump command:

- "Dump command syntax" on page 343
- "Recommendations for performing a backup" on page 346
- "Specifying the backup level" on page 348
- "Understanding increment chains" on page 349
- "Specifying the tape device" on page 351
- "Specifying the dump path" on page 353
- "Specifying a list of files" on page 355
- "Excluding items from a backup" on page 356
- "Specifying a name for a backup" on page 358
- "Specifying no Windows NT ACLs" on page 359
- "Specifying the blocking factor" on page 360
- "Specifying the tape file size" on page 362
- "Specifying updates to the /etc/dumpdates file" on page 363

Dump command syntax

Syntax of the dump command

The `dump` command syntax is as follows:

`dump options arguments dump_path`

Note

Not all options are mandatory, and some do not have any arguments.

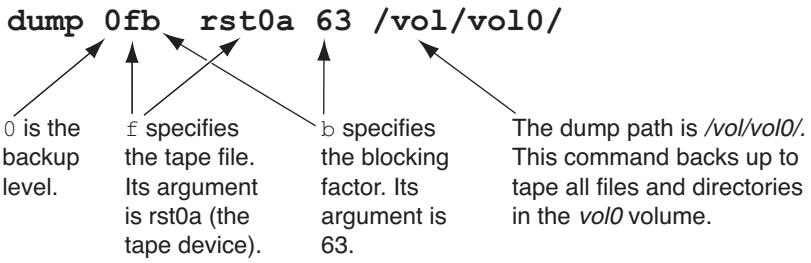
Rules for entering the dump command

The following list describes the rules for entering the `dump` command:

- *options* can be one option or a list of options. You must list all options together; do not separate the options by commas or spaces.
- You can list options in any order.
- You must include a backup level and a tape file in the options.
- *arguments* can be one argument or a list of arguments, each of which is associated with an option.
- List all arguments in the same order as their corresponding options.
- Separate each argument with one or more spaces.
- If the argument is a list, use commas to separate the items in the list.
- *dump_path* is the complete path name of the volume, directory, or qtree batch file to be backed up by the `dump` command.
- Always precede the volume name by `/vol/` even if the volume is a root volume, because between different levels of backups, you might have changed the root volume.

Example of a simple dump command

The following figure illustrates the `dump` command syntax:



Dump options

The following table summarizes the meanings of the `dump` command options. For general recommendations on using them, read "Recommendations for performing a backup" on page 346.

Option	Meaning	For more information, see...
backup level	Level 0 is a full backup; levels 1 through 9 are for incremental backups.	"Specifying the backup level" on page 348 and "Understanding increment chains" on page 349
`A`	Do not back up Windows NT ACLs.	"Specifying no Windows NT ACLs" on page 359
`b`	The blocking factor. **Argument:** The number of 1-KB blocks in each write. For a filer, the range is 4 through 64, and the default is 63.	"Specifying the blocking factor" on page 360
`B`	Write the specified number of tape blocks to a tape file before starting a new tape file. **Argument:** The number of tape blocks in a tape file.	"Specifying the tape file size" on page 362

Option	Meaning	For more information, see...
f	Specify the tape device for the backup. It is mandatory. **Argument:** At least one tape device name as an argument. Separate additional tape device names with commas.	"Specifying the tape device" on page 351
l	Back up only specific files and directories in the dump path. You must use the n option when using the l option.	"Specifying a list of files" on page 355
n	Provide a name for the backup to be recorded in the /etc/dumpdates file. It takes a string as an argument. It is required if you use the l option.	"Specifying a name for a backup" on page 358
Q	Back up all data in the specified volume that does not reside in a qtree.	"Excluding all data in qtrees" on page 356
u	Update the /etc/dumpdates file. You must use this option if you plan to perform incremental backups in the future.	"Specifying updates to the /etc/dumpdates file" on page 363
X	Exclude specified files from the backup. It takes a string specifying exclusion prefixes or suffixes as an argument.	"Excluding specified files" on page 357

Recommendations for performing a backup

General recommendations

Avoid backing up too much data in a single `dump` command to reduce the risk of encountering an error that requires restarting a `dump` command. If the `dump` command encounters an error, however, you might be able to correct the error and proceed from the point where the command terminated. (See "Restarting a dump command backup" on page 382.)

If the filer console appears to be hung during a backup, it is because a backup can take a long time. The console becomes responsive and the prompt returns when the backup is completed.

Recommendations for minimizing backup time and data loss

You can minimize both the time required to perform a backup and the possibility of data loss. The shorter the time for the `dump` command to finish, the more incremental backups you can perform.

- Perform frequent incremental backups to minimize the amount of unrecoverable data in case of errors.

Note

There is a disadvantage to having a large number of incremental backups. When you restore data, you must restore from all the incremental backup tapes, which requires running multiple restores and manipulating multiple tape sets.

- Use local tape drives. The filer can write faster to a local tape drive than to a tape drive attached to a remote system.
- Organize data to be backed up.

  The `dump` command runs faster if the dump path specifies the following:

  - A full volume
  - Full qtrees
  - A full volume excluding qtrees

Recommendations for decreasing tape backup time

- Divide large volumes into smaller volumes or qtrees.

  For example, if you divide a 500-GB volume into three qtrees, you can back up each qtree to a separate tape drive or run separate full backups on three different nights.

- Limit the amount of data in a volume or qtree to be backed up to 200 GB.
- Schedule the backups in appropriate rotations.
- Schedule backups when the load on the filer is light.
- Do not divide a backup into more than 15 qtrees.

Recommendations for minimizing the number of tape drives required

Because Data ONTAP supports the `rmt` protocol, several filers can share the same tape drive. Attach the tape drive to the filer with the most data to back up. Follow these guidelines if multiple filers back up to the same tape drive:

- Use a private network for the backup so that the traffic load on the network does not slow down the backup process.
- Schedule the `dump` command on each filer so that it starts only when no other filers are using the tape drive.

Recommendations for recording information for restores

For ease of use during a restore, you should label the backup tapes with the following information:

- The dump path of each backup on a tape.
- The level of each backup on a tape.
- The date of each backup.
- The blocking factor. This must match for backups and restores.
- Tape file contents of a multifile tape. A brief description of the contents of each tape file on a multifile tape helps you locate a desired tape file for restoring.
- The sequence of tape files on a multifile tape. This enables you to specify which file to restore. To specify a tape file, you need to know where in the sequence of tape files it is.
- The Data ONTAP version of each backup.

Recommendation for verifying backups

If you suspect that a backup might have errors, verify the backup by performing a test restore, as described in “Specifying a test restore” on page 415. For information about restoring information from tape backups, see “Data Restoration from Tape” on page 385.

Specifying the backup level

What a backup level is for

You use a backup level to specify whether all files or only the most recently changed files are backed up.

Kinds of backup levels

A level-0 backup is a full backup. A full backup backs up all the data in the dump path.

Backups at levels from 1 through 9 are incremental backups. An incremental backup backs up only the items in the dump path that have been created or changed since the most recent backup of a lower level.

Specifying a backup level

To specify the backup level, complete the following step.

Step	Action
1	Include the level number as an option. The range is 0 through 9.

Example: The following command performs a level-0 backup of the /vol/vol0 volume:

```
dump 0f rst0a /vol/vol0
```

0 specifies a level 0 backup.

Note

Incremental updates do not run unless the baseline transfer has updated the dumpdates file.

Understanding increment chains

Definition of an increment chain

An increment chain consists of a series of incremental backups of the same path. Because you can specify any level of backup at any time, you must understand increment chains to be able to perform backups that result in the files that you want being restored.

Types of increment chains

There are two types of increment chains:

- A consecutive increment chain is a consecutive sequence of incremental backups that starts with level 0 and is incremented by one. If this is how you do incremental backups, you do not need to read the rest of this section.
- A nonconsecutive increment chain is one where incremental backups skip levels or have levels that are out of sequence. An example of such a chain is one where the levels have been specified in the following order: 0, 2, 3, 1, 4, or more commonly, 0,1,1,1 or 0,1,2,1,2.

Incremental backup sources

Incremental backups base themselves on the most recent backup with a level lower than the current backup as the base of the backup. For example, the sequence of backup levels 0, 2, 3, 1, 4 gives two increment chains: 0, 2, 3 and 0, 1, 4. The bases of the incremental backups are shown in the following table.

Back-up order	Incre-ment level	Incre-ment chain	Base	Files backed up
1	0	Both	Files on the filer	All files in the backup path
2	2	0, 2, 3	The level-0 backup	Files in the backup path created since the level-0 backup
3	3	0, 2, 3	The level-2 backup	Files in the backup path created since the level-2 backup

Back-up order	Increment level	Increment chain	Base	Files backed up
4	1	0, 1, 4	The level-0 backup, because that is the most recent level that is lower than the level-1 backup	Files in the backup path created since the level-0 backup, including files that are in the level-2 and level-3 backups
5	4	0, 1, 4	The level-1 backup, because it is both of a lower level and more recent than the level-0, level-2, or level-3 backups	Files created since the level-1 backup

Specifying the tape device

About specifying tape devices

You must specify at least one tape device to do the backup. If you specify more than one tape device, each tape device in the list is used in the order listed to write a tape file. If the backup requires more tape files than the number of tape devices specified, the last tape device is used for all remaining tape files.

Caution

If you specify more than one rewind device on the same tape drive, the filer displays a warning and terminates the `dump` command.

Note

Filer device names might not be valid on remote tape drive hosts. For tape drives attached to remote hosts, use tape device names that follow the host naming conventions.

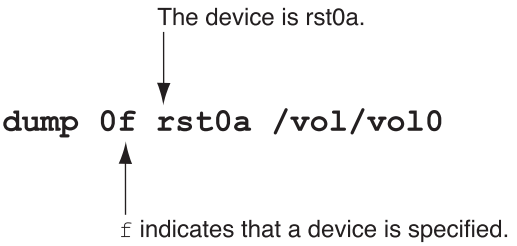
You can specify two kinds of tape devices: local and remote.

Specifying a local tape device

To specify local tape devices for a backup, complete the following step.

Step	Action
1	Use the `f` option with a list of one or more tape devices, separated by commas, as an argument. **Note** You cannot combine local and remote tape devices in one command, and you can write to only one remote machine in a command.

Example: The following command specifies that one tape file be written with one device:

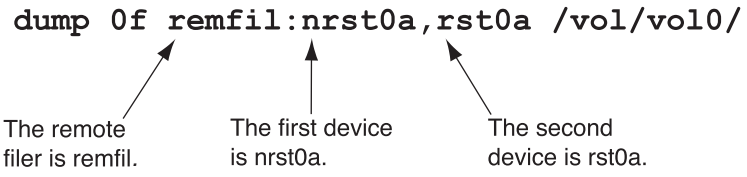


Specifying a tape device on a remote filer

To specify tape devices for a backup on a tape drive attached to a remote filer, complete the following step.

Step	Action
1	Use the f option with a list of one or more tape devices, separated by commas, as an argument. Do not repeat the remote machine name. **Note** You cannot combine local and remote tape devices in one command, and you can write to only one remote machine in a command.

Example: The following command specifies that two tape files, one using nrst0a and the other using rst0a, be written on the remote filer named remfil:



Specifying the dump path

What the dump path does

The dump path specifies one volume, qtree, or subtree to back up. A subtree is a directory in a volume or qtree.

Ways of specifying a dump path

You can specify a dump path in the following ways:

- By specifying a single path name
- By using a qtree batch file

Specifying a single path name

You can simply specify a volume, qtree, or subtree to back up all the data in it. The volume, qtree, or subtree can be in either of the following locations:

- The active file system, for example, /vol/volname/home
- A snapshot, for example, /vol/volname/.snapshot/weekly.0/home

To specify a single dump path, complete the following step.

Step	Action
1	Put the path name of the volume, qtree, or subtree that you want to back up at the end of the `dump` command.

Example: The following command contains the dump path /vol/vol0:

```
dump 0f rst0a /vol/vol0
```

The dump path is */vol/vol0.*

Using a qtree batch file

You can back up several qtrees from a volume with the same command by using a qtree batch file. Using a qtree batch file involves creating the batch file, then specifying it as part of the `dump` command.

Creating a qtree batch file

To create a qtree batch file, complete the following steps.

Step	Action
1	Create a file anywhere in a volume that contains qtrees that you want to back up.
2	Add to the file the name of each qtree that you want to back up, one qtree name to a line. The following rules apply to the batch file: ◆ The qtrees must all be in the same volume as the batch file. ◆ You can have up to 255 qtree names in the file. ◆ There must be no blank lines in the file. **Note** Leading and trailing spaces are treated as part of the qtree name.

Example: The following misc/files/qtreefile qtree batch file in the /vol/vol0 volume backs up the qtrees /vol/vol0/qtree1, /vol/vol0/qtree2, and /vol/vol0/qtree3:

```
qtree1
qtree2
qtree3
```

Specifying a qtree batch file

To specify a qtree batch file, complete the following step.

Step	Action
1	Put the path name of the qtree batch file at the end of the `dump` command.

Example: The following command backs up the qtrees named in the misc/files/qtreefile batch file in the /vol/vol0 volume:

```
dump 0f rst0a /vol/vol0/misc/files/qtreefile
```



Specifying a list of files

About specifying a list of files

You can specify for backup a list of one or more files. The files must all be in the same dump path.

You use this method to back up some, but not all, subdirectories or files in the dump path with only one `dump` command. Using a list is easier than using a `dump` command for each subdirectory or file, and it is quicker because it avoids the overhead of starting up multiple `dump` commands.

Note

Because the `dump` command reads the files from standard input, you must use a script to specify a list of files if you use a Remote Shell connection.

Specifying the file list

To specify a list of files, complete the following steps.

Step	Action
1	Use the `n` and `l` options in the command line.
2	Include a name for the backup as an argument to the `n` option.
3	Put the path name of the volume, qtree, or subtree that you want to back up at the end of the `dump` command.
4	Enter the `dump` command line.
5	In response to prompts, enter each name as a path name relative to the dump path in the `dump` command. **Note** Do not specify a parent directory (..) or a directory that is a symbolic link.
6	To end the list, press the Enter key.

Excluding items from a backup

What you can exclude from a backup

You can exclude the following items from a backup:

- Data in qtrees
- Specified files

About excluding all data in qtrees

You can back up all data in a specified volume that is not in a qtree. The specified volume is the dump path. You use this method if you back up on a qtree basis and want to back up the remaining data in a volume. Usually, the data in qtrees changes frequently, while the remaining data, such as configuration files, changes rarely.

Note

You cannot do incremental backups using this method.

Excluding all data in qtrees

To back up all non-qtree data in a specified volume, complete the following steps.

Step	Action
1	Use the Q option in the command line.

Example: The following command backs up all items in /vol/vol0 that are not in a qtree:

```
dump 0fQ rst0a /vol/vol0
```

Q excludes items in qtrees

The dump path is */vol/vol0.*

About excluding specified files

You can exclude a list of files and all files that match a specified pattern. You specify a pattern to avoid backing up classes of files that you don't want, such as files that end with .core.

Rules for excluding specified files

The rules for constructing a string for excluding files are as follows:

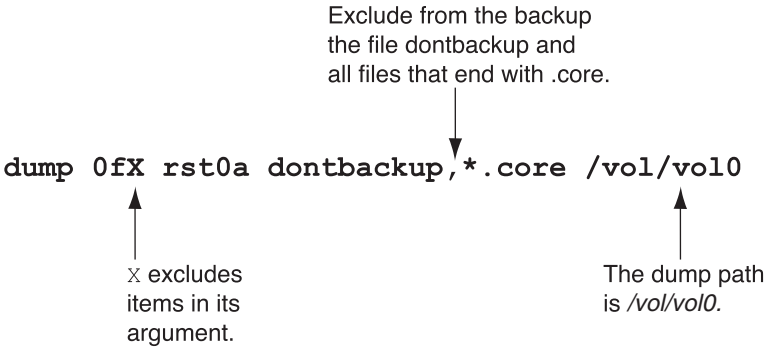
- A string can be a file name.
- You can use the asterisk (*) as a wildcard character.
- The wildcard character must be the first or last character of the string. Each string can contain up to two wildcard characters. For example, you can specify `*.core`, `core.*`, or `*core.*`, but not `core*.1`.
- If you have more than one string, you must separate the strings with a comma.
- You cannot have a comma in the file name or pattern.
- You can specify up to 32 strings.

Excluding specified files

To exclude files from a backup, complete the following steps.

Step	Action
1	Use the `X` option in the command line.
2	Include a string or comma-separated list of strings as an argument to the `X` option.

Example: The following command backs up all items in /vol/vol0 except for the file dontbackup and all files that end in .core:



Specifying a name for a backup

When to specify a name for a backup

You can name a backup and, if you use the `u` option, have the name of the backup recorded in the /etc/dumpdates file. You do this in two situations:

- You are specifying a list of directories or files in the backup with the `l` option.
- You want to monitor the backup history of a backup.

Naming a backup

To specify a name for a backup, complete the following steps.

Step	Action
1	Include the `n` option in the `dump` command line.
2	Include a name for the backup as an argument to the `n` option.

Example: The following command gives the name thisbackup to a backup:

```
dump 0fn rst0a thisbackup /vol/vol0
```

`n` specifies to name this backup.

The name of the backup is thisbackup.

Result: A line like the following appears in the /etc/dumpdates file:

```
thisbackup 0 Tue Jul 24 20:40:09 2001
```

Specifying no Windows NT ACLs

Reasons to omit Windows NT ACLs

You can omit Windows NT ACLs (access control lists) from a backup. This provides a slight performance enhancement. You omit Windows NT ACLs in two situations:

- You plan to restore to a volume that will be used in a UNIX environment or to a pre-5.1 version of Data ONTAP.
- You are backing up no files or directories that contain Windows NT ACLs.

Omitting Windows NT ACLs

To omit Windows NT ACLs from a backup, complete the following step.

Step	Action
1	Include the `A` option in the `dump` command line. **Note** This option does not take an argument.

Example: The following command backs up the /vol/vol0 volume without backing up Windows NT ACLs:

```
dump 0fA rst0a /vol/vol0
```

`A` specifies that no Windows NT ACLs will be backed up.

Specifying the blocking factor

What the blocking factor is

A tape block is 1,024 bytes of data. In the `dump` command, you can specify the number of tape blocks that are transferred in each write operation. This number is called the blocking factor.

Blocking factor range on a filer

If you use a filer for both the backup and the restore, you do not need to specify a blocking factor. Network Appliance recommends that you use the default blocking factor of 63. On a filer, you can specify a blocking factor from 4 through 64.

Blocking factor choice on other machines

On a remote host that is not a filer, you can use a blocking factor from 4 through 256, provided that the host supports the blocking factor that you choose.

Note

SunOS 4.x and earlier versions do not support a blocking factor greater than 63.

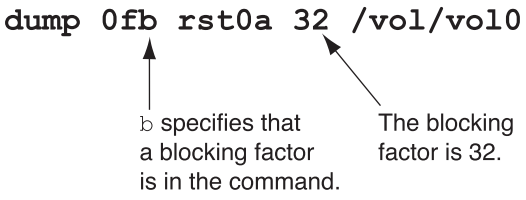
If you plan to restore to a system other than the system that did the backup, the restore system must support the blocking factor that you used for the backup. For example, if you use a blocking factor of 128, the system on which you restore that backup must support a blocking factor of 128. A blocking factor of 128 would mean that you could not restore the backup using a tape drive that is connected to a filer.

Specifying a blocking factor

To specify a blocking factor for a backup, complete the following step.

Step	Action
1	Include the `b` option in the `dump` command line. **Note** This option does not take an argument.

Example: The following command backs up the /vol/vol0 volume using a blocking factor of 32, so that 32 KB of data are transferred at a time:



Specifying the tape file size

Maximum tape file sizes

In the `dump` command, you can specify the maximum size of the tape file in terms of tape blocks. For example, if you want the maximum tape file to be 2 GB, specify 2,097,151. This specifies that the largest tape file can contain 2,097,151 tape blocks, which are 1 kilobyte each. The tape file size must be equal to or greater than the blocking factor; otherwise, the dump process terminates.

When to specify the tape file size

If you do a remote backup or plan to restore the backup on a system other than the filer, you might need to specify a tape file size. Some systems support tape files of a limited size; for example, some Solaris systems do not support tape files larger than 2 GB.

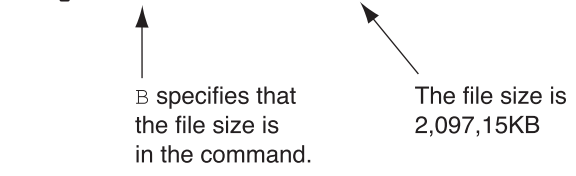
Specifying the tape file size

To specify a tape file size, complete the following steps.

Step	Action
1	Include the `B` option in the `dump` command line.
2	Include the tape file size, in kilobytes, in the `dump` command as an argument to the `B` option. The size applies to all tape files in the backup.

Example: The following command backs up the /vol/vol0 volume using a tape file size of 2,097,151, so that a tape file is no larger than 2 GB:

```
dump 0fB rst0a 2097151 /vol/vol0
```



Specifying updates to the /etc/dumpdates file

What the /etc/dumpdates file is for

The /etc/dumpdates file enables you to keep track of backups. It records the following information:

- The name of the backup. The name is one of the following:
  - If you use the `n` option, the name you supply
  - If you use the `Q` option, the volume you are backing up followed by the notation `/all_non_quota_files`
  - If you use neither, the dump path
- The level of the backup.
- The time of the snapshot used for the backup.

Why to update the /etc/dumpdates file

You update the /etc/dumpdates file for the following reasons:

- You plan to perform incremental backups. The filer uses the data in the /etc/dumpdates file to determine what to include in incremental backups.
- You want to keep a history of a backup.

Principles applying to the /etc/dumpdates file

The following principles apply to the /etc/dumpdates file:

- If the /etc/dumpdates file does not exist when you try to update it, the filer creates it.
- You can edit the /etc/dumpdates file manually, if needed.
- A new backup of the same path and level overwrites the old entry.

Example of an /etc/dumpdates file

An /etc/dumpdates file lists one backup per line. Each line contains the name of the backup, followed by the level of the backup, then the date of the backup. An /etc/dumpdates file looks like the following:

```
/vol/vol1/ 0 Tue Jul 24 22:07:48 2001
/vol/vol0/ 0 Tue Jul 24 21:06:53 2001
/vol/vol0/etc 0 Tue Jul 24 19:06:15 2001
my_named_dump 0 Tue Jul 24 20:40:09 2001
/vol/vol0/all_non_quota_files 0 Tue Jul 24 20:54:06 2001
/vol/vol0/home 0 Tue Jul 24 21:06:39 2001
```

```
/vol/vol1/ 1 Tue Jul 24 22:08:09 2001
/vol/vol1/ 2 Tue Jul 24 22:08:20 2001
my_named_dump 1 Tue Jul 24 22:12:26 2001
/vol/vol0/home 5 Tue Jul 24 22:12:45 2001
```

Updating the /etc/dumpdates file

To update the /etc/dumpdates file, complete the following step.

Step	Action
1	Include the `u` option in the `dump` command line. **Note** This option does not take an argument.

Example: The following command backs up the /vol/vol0 volume and adds the backup information to the /etc/dumpdates file:

```
dump 0fu rst0a /vol/vol0
```

`u` updates the */etc/dumpdates* file.

If the command is issued on Monday, April 16, 2001, at 45 seconds after 1:12 p.m., the following line is added to the /etc/dumpdates file:

```
/vol/vol0/ 0 Mon Apr 16 13:12:45 2001
```

Appending backups to a tape

Why to append backups

If you are backing up small volumes, qtrees, or files, you might want to put several backups on one tape to conserve tapes. Also, adding each backup to the tape after the last backup ensures that backups are sequential.

Appending a backup to a tape

To append a backup to a tape, complete the following steps.

Step	Action
1	Move the tape to the desired location using the `mt` command. See "Controlling tape drives" on page 316.
2	Execute a `dump` command. **Caution** Use no-rewind device names to make sure that the tape is not rewound and that previous backups are not overwritten.

Examples of the dump command

About the examples

Examples of how to perform the following tasks can help you use the `dump` command.

For detailed information

You can use the following examples of the `dump` command to understand the `dump` command and as boilerplate for performing backups.

- "Level-0 backups to a local tape drive" on page 367
- "Incremental backups to a local tape drive" on page 369
- "Backups to a remote tape drive" on page 370
- "Backups to multiple tapes" on page 372
- "Backup of a directory from a snapshot" on page 374
- "Backup of a list of files or directories" on page 375
- "Backup data without ACLs" on page 376
- "Backup specifying a blocking factor" on page 377
- "Backup excluding specified files" on page 378

Level-0 backups to a local tape drive

Examples of level-0 backups

This section provides examples of the following kinds of level-0 backups:

- A full backup
- A full backup with a record
- A full backup minus qtrees

Full backup

The following command performs a full backup of the /vol/vol1/users/tom/specs directory. After the `dump` command is finished, the tape drive rewinds the tape.

```
dump 0f rst0a /vol/vol1/users/tom/specs
```

The elements of this command line are described in the following table.

Element	Meaning
`0`	Do a full backup.
`f`	A tape device is supplied in the command line.
`rst0a`	The tape drive rewinds.
`/vol/vol1/users/tom/specs`	This is the directory to be backed up.

Full backup with a record

The following command performs a full backup of the /vol/vol1/users/tom directory and records the backup in the /etc/dumpdates file. The tape drive rewinds the tape.

```
dump 0uf rst0a /vol/vol1/users/tom
```

The elements of this command line are described in the following table.

Element	Meaning
`0`	Do a full backup.

Element	Meaning
u	Record the backup in the /etc/dumpdates file.
f	A tape device is supplied in the command line.
rst0a	The tape drive rewinds.
/vol/vol1/users/tom	This is the directory to be backed up.

Full backup minus qtrees

The following command performs a full backup of all non-qtree data in the /vol/vol1 volume. The tape drive rewinds the tape.

```
dump 0fQ rst0a /vol/vol1
```

The elements of this command line are described in the following table.

Element	Meaning
0	Do a full backup.
f	A tape device is supplied in the command line.
Q	Do not back up data that is in a qtree.
rst0a	The tape drive rewinds.
/vol/vol1	This is the volume to be backed up.

Incremental backups to a local tape drive

Level-1 backup

The following command performs a level-1 backup of the /vol/vol1 volume to `nrst0a`. This means that only files that have changed since the most recent level-0 backup of the same path are written to tape. After the command is finished, it does not rewind the tape.

```
dump 1uf nrst0a /vol/vol1
```

The elements of this command line are described in the following table.

Element	Meaning
`1`	Do a level-1 backup.
`u`	Record the backup in the /etc/dumpdates file.
`f`	A tape device is supplied in the command line.
`nrst0a`	The tape drive does not rewind.
`/vol/vol1`	This is the volume to be backed up.

Backups to a remote tape drive

Examples of backups to a remote tape drive

This section provides examples of backups to the following kinds of remote tape drives:

- A tape drive attached to a remote filer
- A tape drive attached to a Solaris system
- A tape drive attached to a Solaris system with a 2-GB tape file limit

Tape drive attached to a remote filer

The following command performs a backup to a tape drive attached to a remote filer named netapp1. The tape drive does not rewind the tape.

```
dump 0f netapp1:nrst0a /vol/vol1
```

The elements of this command line are described in the following table.

Element	Meaning
`0`	Do a full backup.
`f`	A tape device is supplied in the command line.
`netapp1`	This is the name of the filer that the tape drive is attached to.
`nrst0a`	The tape drive does not rewind.
`/vol/vol1`	This is the volume to be backed up.

Tape drive attached to a Solaris system

The following command performs a backup to a tape drive on a Solaris system. The tape drive rewinds the tape.

```
dump 0f ritchie:/dev/rmt/0 /vol/vol1
```

The elements of this command line are described in the following table.

Element	Meaning
`0`	Do a full backup.
`f`	A tape device is supplied in the command line.
`ritchie`	This is the name of the Solaris machine to which the tape drive is connected.
`/dev/rmt/0`	This is the name of the tape drive. Tape drive names vary according to the type of Solaris system you use.
`/vol/vol1`	This is the volume to be backed up.

Tape drive attached to a Solaris system with a 2-GB tape file limit

The following command performs a backup to a tape drive on a Solaris system. The thompson system has a 2-GB limit on tape file sizes. The size of the backup is greater than 2 GB but less than 4 GB, so the backup must be broken up into two tape files.

```
dump 0fB thompson:/dev/rmt/0n,/dev/rmt/0n 2097151 /vol/vol1
```

The elements of this command line are described in the following table.

Element	Meaning
`0`	Do a full backup.
`f`	A tape device is supplied in the command line.
`B`	A tape file size maximum is supplied in the command line.
`thompson`	This is the name of the Solaris machine to which the tape drive is connected.
`/dev/rmt/0n`	This is the name of the remote tape drive.
`rst0a`	This is the tape drive rewinds after the second file.
`2097151`	The maximum tape file size is 2 GB.
`/vol/vol1`	This is the volume to be backed up.

Backups to multiple tapes

Examples of backups to multiple tapes

This section provides examples of the following kinds of multiple-tape backups:

- A backup using multiple tapes on single-tape drives
- A backup using multiple tapes on a stacker

Multiple tapes on single-tape drives

The following command backs up the /vol/vol1 volume to the tape on tape drive 0 using `rst0a`. If the backup exceeds the capacity of the tape, the command writes the second tape file on tape drive 1 using `rst1a`. Otherwise, it does not use the tape on `rst1a`.

```
dump 0f rst0a,rst1a /vol/vol1
```

The elements of this command line are described in the following table.

Element	Meaning
`0`	Do a full backup.
`f`	A tape device is supplied in the command line.
`rst0a`	The first tape drive rewinds.
`rst1a`	The second tape device rewinds.
`/vol/vol1`	This is the directory to be backed up.

Multiple tapes on a stacker

The following command backs up the /vol/vol1 volume to the tape on `urst0a`. After the command finishes writing to the first tape drive, it rewinds and unloads the tape. If the volume exceeds the capacity of the tape on `urst0a`, the command writes the second tape file to the same tape drive on `urst0a`. If the command needs to write a third tape file, it prompts you to load a new tape.

```
dump 0f urst0a,urst0a /vol/vol1
```

Caution

If you do not use a stacker when you specify an unload/reload device, the tape drive will unload and reload the same tape, overwriting the tape.

The elements of this command line are described in the following table.

Element	Meaning
`0`	Do a full backup.
`f`	A tape device is supplied in the command line.
`urst0a,urst0a`	The tape stacker rewinds and unloads the first tape. If a second tape is needed, it loads the second tape.
`/vol/vol1`	This is the directory to be backed up.

Backup of a directory from a snapshot

Directory backup from a snapshot

From the weekly snapshot, the following command performs a level-0 backup of the /vol/vol1/home/users directory:

```
dump 0f rst0a /vol/vol1/.snapshot/weekly.0/home/users
```

The elements of this command line are described in the following table.

Element	Meaning
`0`	Do a full backup.
`f`	A tape device is supplied in the command line.
`rst0a`	The tape drive rewinds.
`/vol/vol1/.snapshot/weekly.0/home/users`	This is the directory to be backed up.

Backup of a list of files or directories

List of files or directories

The following example shows the prompts and path name entry when you back up a list of files or directories. The example ends the list of path names with a blank line.

```
dump 0ufnl rst0a user.1.3.5 /vol/vol1/home

DUMP: creating "snapshot_for_backup.0" snapshot.
creating.....................................................
DUMP: Date of this level 0 dump: Tue Jun  4 12:47:14 2001
DUMP: Date of last level 0 dump: Tue May 28 4 12:45:51 2001
DUMP: Dumping /vol/vol0/home to nrst0a
DUMP: mapping (Pass I) [regular files]
DUMP: Reading file names from standard input
user1
user3/jdoe
user5/rroe/src
```

The elements of this command line are described in the following table.

Element	Meaning
`0`	Do a full backup.
`u`	Record the backup in the /etc/dumpdates file.
`f`	A tape device is supplied in the command line.
`n`	A name for the backup is supplied.
`l`	The names of individual files and directories to be backed up are entered interactively.
`rst0a`	The tape drive rewinds.
`user.1.3.5`	This is the name of the backup.
`/vol/vol1/home`	This is the directory that contains the files to be backed up.

Backup data without ACLs

Backup not including ACLs

The following command performs a level-0 backup of the /vol/vol1 volume. The `A` option means that the backup does not include any Windows NT ACL information. Use this option if the files in the volume are for a UNIX-only environment.

```
dump 0uAf rst0a /vol/vol1
```

The elements of this command line are described in the following table.

Element	Meaning
`0`	Do a full backup.
`u`	Record the backup in the /etc/dumpdates file.
`A`	Do not back up Windows NT ACLs.
`f`	A tape device is supplied in the command line.
`rst0a`	The tape drive rewinds.
`/vol/vol1`	This is the volume to be backed up.

Backup specifying a blocking factor

Blocking factor specification

The following command performs a level-0 backup of the /vol/vol1 volume. This command writes 32 KB of data at a time, enabling you to restore the data from systems that limit each write to 32 KB.

```
dump 0ufb rst0a 32 /vol/vol1
```

The elements of this command line are described in the following table.

Element	Meaning
`0`	Do a full backup.
`u`	Record the backup in the /etc/dumpdates file.
`f`	A tape device is supplied in the command line.
`b`	A blocking factor is specified.
`rst0a`	The tape drive rewinds.
`32`	The blocking factor is 32, so write 32 KB of data at a time.
`/vol/vol1`	This is the volume to be backed up.

Backup excluding specified files

File exclusion backup

The following command performs a level-0 backup of the /vol/vol1 volume, but excludes the files that meet certain requirements:

```
dump 0ufX rst0a tmp,*.o,core*,*backup*, /vol/vol1
```

The elements of this command line are described in the following table.

Element	Meaning
`0`	Do a full backup.
`u`	Record the backup in the /etc/dumpdates file.
`f`	A tape device is supplied in the command line.
`X`	An exclude list is specified.
`rst0a`	The tape drive rewinds.
`tmp,*.o,core*,*backup*`	The exclude list specifies files as follows: ◆ `tmp` specifies that the file name is tmp. ◆ `*.o` specifies that the file name ends in `.o` (for example, program.o). ◆ `core*` specifies that the file name begins with the core string (for example, core.small). ◆ `*backup*` specifies that the file name contains the `backup` string (for example, spec.backup.1).
`/vol/vol1`	This is the volume to be backed up.

Verifying the contents of a dump command backup

Reason to verify a backup

You might want to verify a backup initiated by the `dump` command to ensure that all the files you wanted to back up are on the tape.

Verifying a backup

To verify a backup, complete the following steps.

Step	Action
1	From your client, preserve output to the console by using a utility such as `script`.
2	List all the files in a backup by entering the following command: `restore tf rst0a`
3	Compare the list to what you intended to back up.
4	For more detailed verification, use the `N` option of the `restore` command, as described in "Specifying a test restore" on page 415.

Checking the status of a dump command backup

How to find out whether a backup needs to be restarted

To find out whether a backup initiated by the `dump` command is proceeding as expected or has terminated, you can run the `backup status` command. The command provides output in the following format.

```
ID State       Type   Device  Start Date     Level  Path
-- -----       ----   ------  ----------     -----  ----
1  ACTIVE      dump   nrst0a  Nov 28 00:22   0     /vol/vol0
2  ACTIVE      dump   nrst0a  Nov 28 00:22   3     /vol/vol1
4  ACTIVE      NDMP   urst1a  Nov 28 00:22   1     /vol/vol0
6  RESTARTABLE dump           Nov 27 00:22   3     /vol/vol1
```

The elements of this output (the “dump table”) are described in the following table.

Element	Meaning
`ID`	The unique ID assigned to the dump and the index in the software’s internal dump table. As soon as a dump completes, its ID number is deallocated and returned to the pool of available slots. The total number of entries in the dump table is limited to 32
`State`	The state of the dump: ACTIVE or RESTARTABLE
`Type`	The type of dump: standard format (dump) or NDMP protocol (NDMP)
`Device`	The current device to which the dump is writing
`Start Date`	Date on which the backup began
`Level`	The level of the dump (0 through 9)
`Path`	The dump path

How to get details about a specific backup

To get more detailed information about a specific backup initiated by the `dump` command, supply the dump ID at the end of the `backup status` command, for example:

```
filer> backup status 2
State:       ACTIVE                      Type:        dump
Path:        /vol/vol0/src               Level:       0
Options:     b=63, u
Devices:     rst1a,rst2a,rst3a
Completed:   3 tape files
Last Update: Mon Nov 26 00:14:35 2001
```

The `backup status` output for an individual dump ID supplies the following additional information.

Element	Meaning
`Options`	All the options specified for the backup and their respective arguments
`Completed`	The number of tape files that have already been copied
`Last Update`	The time and date of the last completed update

Restarting a dump command backup

Why you might need to restart a backup

You might need to restart a backup initiated by the `dump` command. The `dump` command sometimes does not finish because of such internal or external errors as tape write errors, power outages, accidental user interruptions, or internal inconsistency on the filer.

You might also choose to interrupt and restart a backup to avoid periods of heavy traffic on the filer or to avoid competition for other limited resources on the filer, such as a tape drive. A long backup can be interrupted and restarted later if a more urgent restore (or backup) requires the same tape drive. Restartable backups persist across reboots.

What qualifies as a restartable backup

A failed backup to tape can be restarted if it has completed one tape that has ended in phase IV (file data and ACL data).

How to find out whether a backup needs to be restarted

To find out whether a backup is proceeding as expected or has terminated, you can run the `backup status` command. For more information, see “Checking the status of a dump command backup” on page 380.

How to restart a dump process

To restart a dump process that has been shown to be restartable, complete the following step.

Step	Action
1	From the filer, enter the following command: **`dump R[f` *`comma-separated device-list`*`] {`*`path`*` |`*`ID`*`}`** `f` is an option that enables you to supply a device list. *comma-separated device-list* lets you direct the dump stream to output devices other than those originally designated in the failed dump. A restarted dump process uses this device list the same way a regular dump would. Any device list that is valid to a regular dump will be valid in this case. If a device list is not specified, the command defaults to the remainder of the devices listed but not yet consumed by the failed dump. For example, suppose the following device list was supplied to the previous dump, which failed while writing to `rst2a`: `rst0a,rst1a,rst2a,rst3a,rst4a` The command would use `rst3a,rst4a` as the default. However, if the original device list contained any non-rewinding (`nrst`) devices or any devices not supported by NetApp, users are required to supply a new device list at restart of the dump. *path* is the path that is listed in the dump table (the output of the `backup status` command). If there are multiple entries, that is, entries with exactly the same path, the command prompts you to use the ID to restart the dump. *ID* is the unique ID displayed by the `backup status` command. You can use either *path* or *ID* in most cases. **Result:** The command starts rewriting the dump stream from the beginning of the tape file in which the previous dump was interrupted.

Deleting restartable dump command backups

Deleting restartable backups

To delete a restartable backup, complete the following step.

Step	Action
1	Enter the following command: **`backup terminate ID`** *ID* is the unique ID displayed in the dump table displayed by the `backup status` command. **Result:** The command deletes the specified restartable backup.

How the dump command deletes restartable backups automatically

To prevent restartable backups from accumulating on a filer and taking up unreasonable amounts of disk space, the `dump` command automatically checks the level of use of the snap reserve every 10 minutes. If the snap reserve is over 100%, it deletes the oldest restartable backups until snap reserve usage drops below 100% or there are no more restartable backups to delete.

Data Restoration from Tape 10

About this chapter

Data restoration from tape is the process of copying data files back to the filer from backup tapes that were backed up using the filer `dump` command.

This chapter explains how to use the filer `restore` command for data recovery.

Topics in this chapter

This chapter discusses the following topics:

- "About data restoration" on page 386
- "What the restore command recovers" on page 387
- "How the restore command works" on page 388
- "Preparing to use the restore command" on page 389
- "Using the restore command" on page 393
- "Restoring incremental backups" on page 416
- "Restoring the entire filer" on page 417
- "Restore examples" on page 418

About data restoration

When to restore data from tape

You restore data from tape under the following circumstances:

- Files were deleted from disk but backed up to tape.

  For example, if you inadvertently delete a file and want to restore the file, you can recover the file from tape.
- Files are corrupted. If some files are corrupted, you can restore the subtree containing the files.
- No disk slots are available for expansion. If the filer runs out of storage space, you can do the following tasks:
  - Back up the entire filer.
  - Replace the current disks with disks of greater capacity.
  - Set up volumes on new disks.
  - Restore the filer from tapes.
- The entire filer is damaged and unusable. Call Network Appliance Technical Support to determine whether you can repair the filer and restore data from tape.

When not to restore data from tape

You would not restore a deleted file from tape if you can recover the deleted file from a snapshot on the filer. If the file is in a snapshot, copying the file to the active file system is faster than recovering the file from tape.

Note

If you attempt an incremental restore to a filer running Data ONTAP 6.2 or later from a filer running a version earlier than 6.2, the restore will fail because of a formatting code change between the two code releases. You need to run the full backup again after you have upgraded to 6.2.

What the restore command recovers

What data can be recovered

The `restore` command enables you to recover all the information that you backed up with the `dump` command so that it is the same as when you did the backup.

The `restore` command can recover the following data:

- Contents of files and directories
- UNIX file permissions
- Windows NT access control lists (ACLs)

Note

If you restore a file that has only UNIX file permissions into an NT file system (NTFS) qtree or volume, the file has no Windows NT ACLs. The filer uses only the UNIX file permissions on this file until you create a Windows NT ACL on it.

- Qtree information

Note

Qtree information is used only if a qtree is restored to the root of a volume. If a qtree is restored to a lower directory, such as /vol/vol0/subdir/lowerdir, it ceases to be a qtree and qtree information is not used.

- All other file and directory attributes
- Windows NT streams information

Compatibility

You can restore data from previous versions of Data ONTAP using the `restore` command. However, you cannot do an incremental restore to a filer running Data ONTAP 6.2 or later from a filer running a version earlier than 6.2. You need to run the full backup again after you have upgraded.

How the restore command works

The restore process

The process in the following table shows how the `restore` command works.

Stage	Action
1	Data ONTAP catalogs the files that need to be extracted from the tape.
2	Data ONTAP creates directories and empty files.
3	Data ONTAP reads a file from tape, writes it to disk, and sets the permissions–including ACLs–on it.
4	Data ONTAP repeats steps 2 and 3 until all the specified files are copied from the tape.

Preparing to use the restore command

General preparations

Before restoring data with the `restore` command, you should make sure that you have the required materials, decide where to enter the `restore` command, and prepare the destination for the restore.

For detailed information

The following topics describe in detail how to prepare to restore files using the `restore` command:

- "Getting required materials" on page 390
- "Deciding where to enter the restore command" on page 391
- "Preparing the destination" on page 392

Getting required materials

Required information

Before restoring data with the `restore` command, you must have the following information:

- The level of the restore
- The tape device you used for each tape file in the backup that you are restoring
- The path into which you are restoring the material
- The blocking factor used in the backup

Required tape drives and tapes

You must meet these prerequisites for the `restore` command to run successfully:

- If you are doing an incremental restore, you need all the tapes in the backup chain.
- You need a tape drive that is available and compatible with the tape to be restored from.

Required space on destination filer

You need about 100 MB more space on the destination filer than the amount of data to be restored.

Caution

The `restore` command terminates if it runs out of space.

Deciding where to enter the restore command

Where to enter the command

You can enter the `restore` command through a Remote Shell connection, such as `rsh`, or on the console.

Benefits of entering the restore command through a Remote Shell

Entering the `restore` command through a Remote Shell connection gives you these benefits:

- When the `restore` command is in progress, you can still use the console to manage the filer.
- You can start multiple `restore` commands through a Remote Shell connection if other tape drives are available.
- It is less likely that someone will inadvertently terminate the `restore` command, especially if it is run in the background from a UNIX system. If you enter a `restore` command on the console, it could be terminated by Ctrl-C entered on a host connected to the filer using Telnet.

Benefit of entering the restore command on the console

The benefit of entering the `restore` command on the console is that you can read and respond to screen messages displayed by the command. For example, the command might prompt you for another tape to complete the recovery.

Preparing the destination

About preparing the destination

If you are restoring the backup to its original path, you do not need to prepare the target volume, qtree, or subtree. If you are restoring the backup to a different destination, you must prepare the location.

Preparing the destination

If you are restoring to a location other than the original location, complete the following step.

Step	Action	
1	**If you are restoring...**	**Then...**
	A volume	Create a new volume.
	A qtree or a directory	Rename or move files that are likely to have the same names as files you are restoring. **Caution** If a restored file has the same name as an existing file, the existing file is overwritten by the restored file. Directories are not overwritten.

Using the restore command

About the restore command

The `restore` command enables you to specify four types of restores and to specify several items in a restore. This section describes the principles of the `restore` command, then the kinds of restores you can perform and the additional items that you can specify.

For detailed information

The following topics describe how to use and construct a `restore` command:

- "Restore command principles" on page 394
- "Specifying an individual file restore" on page 398
- "Specifying a full restore" on page 399
- "Specifying a table-of-contents restore" on page 400
- "Specifying a resumed restore" on page 402
- "Specifying tape devices" on page 404
- "Specifying a single tape file on a multifile tape" on page 406
- "Specifying the restore destination" on page 407
- "Specifying the blocking factor" on page 408
- "Specifying detailed status output" on page 409
- "Specifying ignoring inode limitations" on page 410
- "Specifying automatic confirmations" on page 412
- "Specifying no Windows NT ACLs" on page 413
- "Specifying no qtree information" on page 414
- "Specifying a test restore" on page 415

Restore command principles

Syntax of the restore command

The `restore` command syntax is as follows:

`restore` *`options`* `[`*`arguments`*`] [`*`files ...`*`]`

options can be one restore type with modifiers.

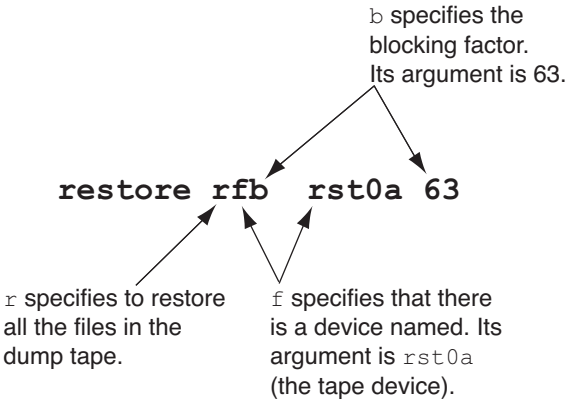
Rules for using the restore command

Follow these rules when you enter the `restore` command:

- Specify one and only one restore type.
- Specify multiple options without intervening spaces.
- Enter the arguments for each option in the order that you specify the options. Separate each argument from the next with a space.
- If the destination for each file is the same as the location from which it was backed up, you do not need to explicitly specify a destination.

Example of a simple restore command

The following command illustrates the `restore` command syntax. It restores all the files in the dump tape to the same destinations from which they were backed up.



Restore types

A restore type specifies the type of restore you are performing. For a restore from tape, you must specify one and only one restore type. The following table summarizes the restore types.

Restore type	Meaning	Option	For more information, see...
Restart	Restarts data recovery after an interruption	`R`	"Specifying a resumed restore" on page 402
Qtree table of contents	Lists qtree names and qtree information in a restore	`T`	"Specifying a table-of-contents restore" on page 400
Full	Rebuilds the file system or subtree. If you are applying incrementals, you must specify this option	`r`	"Specifying a full restore" on page 399
File table of contents	Lists file names in a restore	`t`	"Specifying a table-of-contents restore" on page 400
File	Extracts an individual file or subtree from the backup	`x`	"Specifying an individual file restore" on page 398

Modifiers

Modifiers specify optional actions. The following table summarizes the modifiers.

Option	Meaning	For more information, see...
`A`	Specifies that `restore` does not restore Windows NT ACLs	"Specifying no Windows NT ACLs" on page 413

Option	Meaning	For more information, see...
D	Specifies the directory into which the files are restored **Argument:** The directory into which you are restoring files. Without an argument, the files are restored to the directory from which they were backed up.	"Specifying the restore destination" on page 407
F	Forces `restore` to continue regardless of inode limitations	"Specifying ignoring inode limitations" on page 410
N	Reads backup tapes without writing to the filer	"Specifying a test restore" on page 415
Q	Ignores qtree information	"Specifying no qtree information" on page 414
b	Specifies the blocking factor. **Argument:** The blocking factor that you used in the backup that you are restoring	"Specifying the blocking factor" on page 408
f	Specifies the tape device for each tape file **Argument:** The name of one or more tape devices, separated by commas	"Specifying tape devices" on page 404
s	Specifies the relative position of a tape file if multiple tape files exist on a tape. File numbering starts at 1 from the current tape position **Argument:** The tape file number	"Specifying a single tape file on a multifile tape" on page 406

Option	Meaning	For more information, see...
`v`	Specifies that `restore` displays the inode number of each file restored	"Specifying detailed status output" on page 409
`y`	Specifies that `restore` not ask whether it should terminate when it gets an error	"Specifying automatic confirmations" on page 412

Executing a restore command

Follow these steps to execute a `restore` command.

Step	Action
1	Place the tape containing the first tape file of the backup in the tape drive that you specify
2	Enter the `restore` command
3	If prompted, insert the next tape of the backup that you are restoring into the appropriate tape drive
4	Repeat Step 3 until the restore is complete

Specifying an individual file restore

About restoring an individual file

You can specify the restore of one or more directories or files from a backup.

Specifying individual file restores

To restore an individual directory or file, complete the following steps.

Step	Action
1	Use the `x` option in the `restore` command line.
2	At the end of the command line, include the path names relative to the dump path of the files or directories that you want to restore. Separate path names with a space. **Caution** If you do not have a path in the command line, the `restore` command restores everything on the tape.

Example: The following command restores the /src directory and puts it in the location from which it was backed up:

```
restore xf rst0a /src
```

`X` restores a specified file.

The specified file is the */src* directory.

Specifying a full restore

About full restores

A full restore rebuilds the file system, qtree, or subtree that was in the backup that a tape file contains.

Specifying a full restore

To specify a full restore, complete the following step.

Step	Action
1	Use the `r` option in the `restore` command line.

Example: The following command restores the /src directory and puts it in the same directory that it was backed up from:

```
restore rf rst0a
```

`r` performs a full restore.

Specifying a table-of-contents restore

About displaying a table of contents of a tape file

You can display a table of contents of the files or qtrees in a tape file rather than do an actual data restore. This is useful in determining what files or qtrees are on a tape and their locations. For qtrees, the restore lists qtree properties.

A table-of-contents restore takes much less time than a full restore because only the list of files in the backup is read. However, it uses a lot of CPU time because of the extensive output produced.

Use Remote Shell for a table-of-contents restore

In general, you should run a table-of-contents restore from a Remote Shell connection because an enormous output is generated. Usually, you can control the output more easily when it is sent to a client console rather than to the filer console. Client consoles are also more flexible and enable you to save output.

Furthermore, you rarely need to change tapes with the `t` option. The command needs to read only the directory information from the tape and none of the files or qtrees. Because directory information tends to constitute a small part of a backup, it is almost always located on one tape. Also, table-of-contents restores work with multiple tape files specified on the command line.

Types of table-of-contents restores

You can specify two kinds of tables of contents: file and qtree. You cannot combine the two types in a single command.

Type	Meaning	Option
File	Lists all the file names in a backup. If you specify path names, only the files in the path names are listed.	`t`
Qtree	Lists qtrees and their settings for security style and Windows NT oplocks for all qtrees. If you specify qtree names, the information for only those qtrees is listed if they are in the backup.	`T`

Specifying table-of-contents restores

To specify a table-of-contents restore, complete the following step.

Step	Action
1	Use the `T` or `t` option in the `restore` command line, with files as an argument. If there is no argument, the entire content of a backup is listed.

Example: The following command lists all files in a backup:

```
restore tf rst0a
```

`t` lists all files.

Example: The following command lists all qtree names in a backup:

```
restore Tf rst0a
```

`T` lists all qtree names.

Specifying a resumed restore

About resuming a restore

If an entire tape file restore is stopped, you can resume the restore and avoid restoring again what has already been restored.

Resume restrictions

Resuming a restore has the following restrictions:

- You cannot resume an interrupted table-of-contents restore or an interrupted individual file restore.
- You can resume only restores that you started with the `r` or `R` options.
- You can resume a `restore` command only if the backup consists of multiple tape files.
- You can resume a `restore` command only if the command is for a full restore. If the `restore` command is for extracting an individual file or subtree from a backup (that is, if you use the `x` option), you cannot resume the restore.
- You can resume a restore only if you received a message similar to the following during the restore:

```
RESTORE: Fri Aug 31 22:22:35 2001: Writing data to files.
```

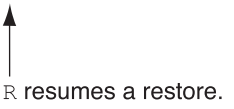
Specifying resuming restores

To specify resuming a restore, complete the following steps.

Step	Action
1	Use the `R` option in the `restore` command line instead of the `r` option as the first option on the command line. It does not take an argument.
2	Enter the rest of the same `restore` command as the one that was interrupted, but include only the tape files that were not restored.
3	Follow the prompts.

Example: The following command resumes a restore.

```
restore Rf rst0a
```



Specifying tape devices

About backup tape devices

When you performed a backup, you specified one or more tape devices. The files written by these devices can be on one or more tapes.

Compression types must be the same

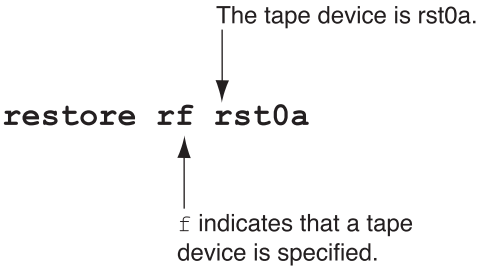
You must use the same compression type to restore a backup as you did to perform the backup; however, you can use a different rewind type and device number. For example, you can use `rst1a` and tape drive 1 to restore a backup done on `nrst0a`, provided that the two tape drives use the same kind of tape.

Specifying the tape devices

To specify the tape devices for restores, complete the following steps.

Step	Action
1	Use the `f` option in the `restore` command line.
2	List the tape devices as an argument to the `f` option in the same order that you used in the backup. Separate multiple tape devices with a comma. The `restore` command restores from tape files consecutively, using the tape devices in the order that they appear in the command line. **Caution** If you do not specify at least one tape device, the `restore` command terminates.

Example: The following command specifies the `rst0a` device for a backup:



Specifying a single tape file on a multifile tape

About multifile tapes

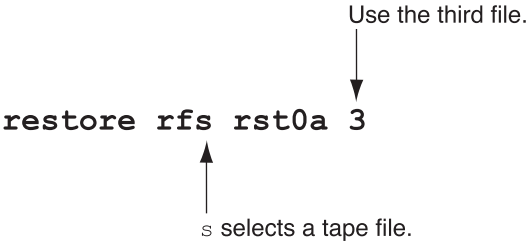
You can have more than one tape file on a tape. Tape files do not have names. You can restore a single tape file on a tape that contains more than one tape file only by moving the tape to the beginning of the file.

Specifying the file

To specify a single tape file to be restored from a multifile tape, complete the following steps.

Step	Action
1	Use the `f` option in the `restore` command line.
2	Use the same tape compression type as an argument to the `f` option that you used in the backup.
3	Use the `s` option in the `restore` command line to select the appropriate backup.
4	Include the relative position of the tape file that you are restoring in the command line as an argument to the `s` option. **Note** Count the relative position from the current tape position. It might be best to rewind the tape and start from the beginning of the tape.

Example: The tape has been rewound, and the following command restores the third tape file from the beginning of the tape, then rewinds the tape:



Specifying the restore destination

When to specify the restore destination

The destination acts as the root of the backup that you are restoring. You specify a different restore destination if you are moving items. For example, if you created a backup and then installed multiple volumes on the filer, when you perform a restore, you might specify a different volume or directory from the original.

If you do not specify a restore destination, the files are restored into the locations from which they were backed up.

Note

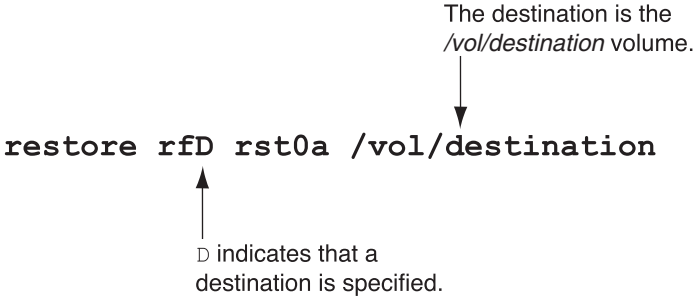
Network Appliance recommends that you specify a restore destination even if you are restoring to the same destination that you back up from. This ensures the files are restored where you want them to go and are traceable to that location.

Specifying the destination

To specify the restore destination, complete the following steps.

Step	Action
1	Use the `D` option in the `restore` command line.
2	Include the absolute path name of the restore destination as an argument to the `D` option.

Example: The following command restores a backup and puts it in the /vol/destination volume:



Specifying the blocking factor

About the blocking factor

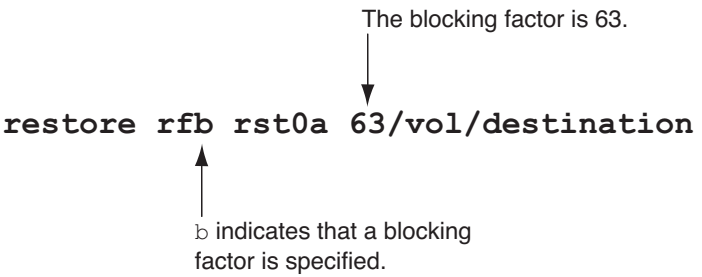
The blocking factor specifies the number of tape blocks that are transferred in each write operation. A tape block is 1 kilobyte of data. You must use the same blocking factor that you used for the backup. The default blocking factor is 63.

Specifying the factor

To specify the blocking factor, complete the following steps.

Step	Action
1	Use the `b` option in the `restore` command line.
2	Include the blocking factor as an argument to that option.

Example: The following command restores a backup and puts it in the /vol/destination volume:



Specifying detailed status output

About detailed status output

You can get information about the progress of a restore on a file-by-file basis. If you have a restore problem, this output can be useful for your own diagnostics, as well as for Network Appliance Technical Support.

Because of the volume of detailed information that needs to be processed by a console, getting detailed output can slow down a restore considerably.

Specifying detailed output

To get status information about each file recovered, complete the following step.

Step	Action
1	Use the `v` option in the `restore` command line. **Note** This option does not take an argument.

Example: The following command restores a backup and produces status information about each file recovered:

```
restore rfv rst0a
```

`v` produces status information about each file recovered.

Specifying ignoring inode limitations

About inode limitations

Inodes are data structures that contain information about files. The number of files, and, therefore, the number of inodes per volume, is determined by the `maxfiles` command. For information about setting the maximum number of files per volume and displaying inode information, see the *System Administrator's Guide*.

Reasons to ignore inode limitations

The `restore` command assumes that the files being restored are added to the number of files on the filer, and, therefore, that the inodes are added to the filer. When the total of inodes in the restore and the inodes on a filer exceeds the number of inodes that are allowed on a filer, the restore is terminated.

However, if a restore updates an existing file, the inode count remains the same. Therefore, if you are sure that the restore consists mostly of files to be updated rather than new files, you can safely specify that the filer ignore the calculations of the `restore` command.

Caution

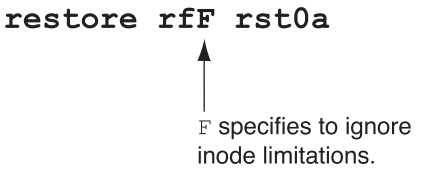
If a restore causes the number of inodes to exceed the maximum allowed, the restore is terminated.

Ignoring inode limitations

To specify a restore that ignores inode limitations, complete the following step.

Step	Action
1	Use the `F` option in the `restore` command line. **Note** This option does not take an argument.

Example: The following command restores a backup and ignores inode limitations:



Specifying automatic confirmations

About automatic confirmations

Automatic confirmation means to answer *yes* to any restore questions.

You usually use this mode on restores run using a Remote Shell connection. A Remote Shell connection does not let you interact with the `restore` command; therefore, if the `restore` command requires user input and is run using a Remote Shell connection, it usually terminates. Specifying confirmation mode enables restores to be completed in most cases. Even with the `y` option, however, the `restore` command does not complete restores if it encounters hard media errors or unclean drives. In this case, it fails and exits.

Caution

This option is not recommended for critical restores because it can cause silent failure.

Specifying automatic confirmations

To specify automatic confirmations, complete the following step.

Step	Action
1	Use the `y` option in the `restore` command line. **Note** This option does not take an argument.

Example: The following command restores a backup with automatic confirmations:

```
restore rfy rst0a
```

`y` specifies automatic confirmations.

Specifying no Windows NT ACLs

About omitting Windows NT ACLs

You can omit Windows NT ACLs from a restore. This provides a slight performance enhancement. You omit Windows NT ACLs in two situations:

- You plan to restore to a UNIX-only environment or to a pre-5.1 version of Data ONTAP.
- Your backup has no files or directories that contain Windows NT ACLs.

Omitting Windows NT ACLs

To omit Windows NT ACLs from a restore, complete the following step.

Step	Action
1	Include the `A` option in the `restore` command line. **Note** This option does not take an argument.

Example: The following command restores a backup but does not restore Windows NT ACLs:

```
restore rfA rst0a
```

`A` specifies not to restore Windows NT ACLs.

Specifying no qtree information

About omitting qtree information

You can omit qtree information from a restore. Qtrees are then restored as ordinary directories. You specify no qtree information in a restore if you want to restore qtrees as regular directories.

Omitting qtree information

To omit qtree information from a restore, complete the following step.

Step	Action
1	Include the `Q` option in the `restore` command line. **Note** This option does not take an argument.

Example: The following command restores a backup but does not restore qtree information:

```
restore rfQ rst0a
```

`Q` specifies not to restore qtree information.

Specifying a test restore

About test restores

You can test a restore by performing a restore that reads the tape but does not write to the filer.

Note
Because a test restore is dependent on the speed of reading from tape, it takes almost as much time as an actual restore.

When to test a restore

You would do a test restore in the following situations:

- To verify a backup tape that is old and might have deteriorated
- To verify that the set of tapes you have is complete
- To verify a backup tape that you believe was not written properly
- To quickly make sure that a block size works if it the block size is unknown

Specifying a test restore

To specify a test restore, complete the following step.

Step	Action
1	Include the `N` option in the `restore` command line. **Note** This option does not take an argument.

Example: The following command performs a test restore of a backup:

```
restore rfN rst0a
```

`N` specifies a test restore.

Restoring incremental backups

Required tapes

Incremental restores build on each other the way incremental backups build on the initial level-0 backup. Therefore, to restore an incremental backup, you need all the backup tapes from the level-0 backup through the last backup that you want to restore.

For additional information about incremental backups and backup chains, see "Specifying the backup level" on page 348 and "Understanding increment chains" on page 349.

Restoring an incremental backup

To restore an incremental backup, complete the following steps.

Step	Action
1	Restore the level-0 backup.
2	Follow the prompts. You might be asked to remove or insert tapes.
3	Restore each incremental backup in the increment chain that you want to restore, starting with the lowest-level backup and going to the last backup that you want to restore. **Note** During an incremental restore operation, a temporary directory labeled `.restore_do_not_touch_`*xxxxxxx* will appear in the active file system. Do not edit or delete this directory. The system will delete this directory after the current incremental restore operation is completed.
4	From the client, remove the restore_symboltable file in the root of the destination directory.

Restoring the entire filer

About restoring the entire filer

You can restore a filer only to disks that you have initialized.

How you restore an entire filer depends on whether you backed up each volume in one backup or as individual directories and qtrees.

If there is one backup for each volume

To restore an entire filer if you used one `dump` command to back up each volume, complete the following step.

Step	Action
1	Restore each backup.

If each volume was backed up as subtrees or qtrees

To restore an entire filer if you used separate `dump` commands to back up files, directories, and qtrees that make up each volume, complete the following steps.

Step	Action
1	Create the desired volumes.
2	Restore each backup to the appropriate volume.

Restore examples

About the examples

Examples of how to perform the following tasks can help you use the `restore` command.

For detailed information

You can use the following examples of the `restore` command to understand the `restore` command and as boilerplate for performing restores:

- "Restoring an entire backup" on page 419
- "Restoring individual files or directories" on page 421
- "Restoring using a remote tape drive" on page 422
- "Multiple tape restores" on page 423

About prompts

In the `restore` command prompts, the following terms are used interchangeably to mean tape file or tape cartridge:

- Tape name
- Tape volume
- Tape

Restoring an entire backup

Restoring a backup to its original location

Backup: `dump 0f rst0a /vol/vol1/`

Restore: `restore rfD rst0a /vol/vol1/`

This command restores the entire tape file to its original location, in this case /vol/vol1. The tape drive rewinds the tape.

The elements of this command line are described in the following table.

Element	Meaning
`r`	Do a full restore.
`f`	A tape device is supplied in the command line.
`D`	The target directory is specified.
`rst0a`	The restore is done using the `rst0a` tape device.

Restoring a backup to a different location

Backup: `dump 0f rst0a /vol/vol1/users/tom`

Restore: `restore rfD rst0a /vol/vol1/home/tom`

This command restores the /vol/vol1/users/tom directory to the /vol/vol1/home/tom directory. The tape drive rewinds the tape.

The elements of this command line are described in the following table.

Element	Meaning
`r`	Do a full restore.
`f`	A tape device is supplied in the command line.
`D`	The target directory is specified.
`rst0a`	The restore is done using the `rst0a` tape device.

Element	Meaning
`/vol/vol1/home/tom`	The restored data is put in the /vol/vol1/home/tom directory.

Restoring individual files or directories

About restoring a single file

The `restore` command can extract individual qtrees, subdirectories, or data files from a backup that contains many qtrees, ordinary directories, or data files.

Restoring individual files

Backup: `dump 0f rst0a /vol/vol1/users/tom`

Restore: `restore xfD rst0a /vol/vol1/u1/tom spec4 spec5`

This command extracts the /vol/vol1/users/tom/spec4 and /vol/vol1/users/tom/spec5 files from the /vol/vol1/users/tom directory and places them in the /vol/vol1/u1/tom directory. The tape drive rewinds the tape.

The elements of this command line are described in the following table.

Element	Meaning
`x`	Restore a single file or directory.
`f`	A tape device is supplied in the command line.
`D`	Restore to a different location from the original.
`rst0a`	The restore is done using the `rst0a` tape device.
`/vol/vol1/u1/tom`	The restored data is put in the /vol/vol1/u1/tom directory.
`spec4 spec5`	The /vol/vol1/users/tom/spec4 and /vol/vol1/users/tom/spec5 files are extracted.

Restoring using a remote tape drive

Filer restore using a tape drive attached to a remote filer

Backup: `dump 0f netapp1:rst0a /vol/vol1`

Restore: `restore rf netapp1:rst0a`

This command performs a restore from a tape drive attached to a remote filer named netapp1. The tape drive then rewinds the tape.

The elements of this command line are described in the following table.

Element	Meaning
`r`	Do a full restore.
`f`	A tape device is supplied in the command line.
`netapp1`	This is the filer name.
`rst0a`	The restore is done using the `rst0a` tape device.

Filer restore using a tape drive attached to a Solaris system

Backup: `dump 0f ritchie:/dev/rmt/0 /vol/vol1`

Restore: `restore rf ritchie:/dev/rmt/0`

This command performs a restore from a tape drive on a Solaris system.

The elements of this command line are described in the following table.

Element	Meaning
`r`	Do a full restore.
`f`	A tape device is supplied in the command line.
`ritchie`	This is the name of the Solaris machine to which the tape drive is connected.
`/dev/rmt/0`	This is the name of the tape device.

Multiple tape restores

Examples of multiple tape restores

This section provides examples of the following kinds of multiple tape restores:

- Multiple tapes on one single-tape drive
- Multiple tapes on two single-tape drives
- Multiple tapes on a tape stacker

Multiple tapes on one single-tape tape drive

Backup: `dump 0f rst0a /vol/vol1`

Restore: `restore rf rst0a`

This command restores the /vol/vol1 volume from the two tapes it took to back it up. You are prompted for the next tape when the first tape is restored.

The elements of this command line are described in the following table.

Element	Meaning
`r`	Do a full restore.
`f`	A tape device is supplied in the command line.
`rst0a`	The restore is done using the `rst0a` tape device; the `restore` command prompts for the second tape.

Multiple tapes on two single-tape tape drives

Backup: `dump 0f rst0a,rst1a /vol/vol1`

Restore: `restore rf rst0a,rst1a`

The first tape is in tape drive 0 and the second tape is in tape drive 1. This command restores the /vol/vol1 volume from the two tapes it took to back it up. It uses the tape in the second tape drive when the first tape is restored.

The elements of this command line are described in the following table.

Element	Meaning
`r`	Do a full restore.
`f`	A tape device is supplied in the command line.
`rst0a`	The restore is done using the `rst0a` tape device for the first tape
`rst1a`	The restore is done using the `rst1a` tape device for the second tape.

Multiple tapes on a tape stacker

Backup: `dump 0f urst0a,urst0a /vol/vol1`

Restore: `restore rf urst0a,urst0a /vol/vol1`

This command restores the /vol/vol1 volume from the two tapes it took to back it up. It unloads the first tape and loads the second tape.

The elements of this command line are described in the following table.

Element	Meaning
`r`	Do a full backup.
`f`	A tape device is supplied in the command line.
`urst0a,urst0a`	The tape drive unloads and loads each tape.

Using NDMP Services 11

About this chapter

This chapter describes how you can use Network Data Management Protocol (NDMP) services on your filer to enable network-based backup and recovery using a number of NDMP-enabled commercial backup applications. It also describes how you can monitor NDMP services running on the filer and how you can use `ndmpcopy` to migrate data efficiently within or between filers.

Topics in this chapter

This chapter discusses the following topics:

- "Learning about NDMP services on the filer" on page 426
- "Using tape devices with NDMP" on page 429
- "Preparing a filer for basic NDMP backup application management" on page 431
- "Enabling or disabling NDMP service" on page 433
- "Managing NDMP security features" on page 434
- "Specifying the NDMP version" on page 439
- "Specifying a preferred network interface" on page 441
- "Displaying NDMP session status" on page 443
- "Displaying NDMP debug messages" on page 446
- "Terminating an NDMP session" on page 449
- "Copying data with ndmpcopy" on page 450

Learning about NDMP services on the filer

What NDMP is

The Network Data Management Protocol (NDMP) is a standardized protocol for controlling backup, recovery, and other transfers of data between primary and secondary storage devices such as filers and tape libraries.

By enabling NDMP protocol support on a filer, you enable that filer to carry out communications with NDMP-enabled commercial network-attached backup applications (also called Data Management Applications or DMAs), data servers, and tape servers participating in backup or recovery operations.

NDMP also provides low-level control of tape devices and medium changers.

Advantages of NDMP

Data ONTAP NDMP backup and recovery operations use the same dump and restore services described in the chapters about data backup to tape and data restoration from tape. However, accessing these data protection services through backup applications that support NDMP offers a number of advantages.

- NDMP backup applications provide sophisticated scheduling of data protection operations across multiple filers. They also provide medium management and tape inventory management services to eliminate or minimize manual tape handling during data protection operations.
- NDMP backup applications support data cataloging services that simplify the process of locating specific recovery data. Direct Access Recovery (DAR) optimizes the access of specific data from large backup tape sets.
- NDMP supports multiple topology configurations, allowing efficient sharing of secondary storage (tape library) resources through the use of three-way network data connections. NDMP backup applications typically provide user-friendly interfaces that simplify the management of data protection services.

Some common NDMP tape backup topologies

NDMP supports a number of topologies and configurations between backup applications and filers or other NDMP servers providing data (file systems) and tape services.

Filer-to-local-tape: In the simplest configuration, a backup application backs up data from a filer to a tape subsystem attached to the filer. The NDMP control connection exists across the network boundary. The NDMP data connection that exists within the filer between the data and tape services is called an NDMP local configuration.

Filer-to-filer-to-tape: A backup application can also back up data from a filer to a tape library (a medium changer with one or more tape drives) attached to another filer. In this case, the NDMP data connection between the data and tape services is provided by a TCP/IP network connection. This is referred to as an NDMP three-way filer-to-filer configuration.

Filer-to-network attached tape library: NDMP-enabled tape libraries provide a variation of the three-way configuration. In this case, the tape library attaches directly to the TCP/IP network and communicates with the backup application and the filer through an internal NDMP server.

Filer-to-data-server-to-tape or data-server-to-filer-to-tape: NDMP also supports filer-to-data-server and data-server-to-filer three-way configurations, although these variants are less widely deployed. Filer-to-server allows filer data to be backed up to a tape library attached to the backup application host or to another data server system. The server-to-filer configuration allows server data to be backed up to a filer-attached tape library.

NDMP security

Data ONTAP provides features for preventing or monitoring for unauthorized use of NDMP connections to your filer. You can restrict the set of backup application hosts permitted to start NDMP sessions on a filer. You can specify the authentication method to use (text or challenge) in order to allow NDMP requests. You can enable or disable monitoring of NDMP connection requests.

Data ONTAP also generates an NDMP-specific passwords for administers who do not have root privilege on the target filer.

Considerations when using NDMP

Take the following considerations into account when starting NDMP service on your filer:

- Data ONTAP supports a maximum of eight concurrent backup and/or restore operations. The maximum includes backups initiated by NDMP as well as by the filer's `dump` and/or `restore` command.
- NDMP backup applications require specification of a target system password. To enable successful authentication by NDMP services on the

filer, you must use either the filer's root password or, to authenticate a non-root user administrator, a system-generated NDMP-specific password.

- NDMP services optionally generate file history data at the request of NDMP backup applications. File history is used by backup applications to enable optimized recovery of selected subsets of data from a backup image. File history generation and processing is time-consuming and CPU-intensive for both the filer and the backup application. If your data protection needs are limited to disaster recovery, where the entire backup image will be recovered, you may want to disable file history generation to reduce backup times. See your backup application documentation to determine if it is possible to disable NDMP file history generation.
- NDMP does not support Unicode format. If you use a third-party tool to back up a filer using the Unicode directory format, the NDMP primary storage displays the file names in the NDMP file history database in two ways:
  - If possible, the filer replaces the Unicode format with the NFS character set.
  - If the filer cannot replace the Unicode format with the NFS character set, the NDMP client displays the file names as: *XXXXXXX*, where *XXXXXXX* is a character string.
- Because NDMP uses the `restore` command to recover files, you can use NDMP to restore files with correct file names regardless of how the client displays the file names in the NDMP file history database.

Using tape devices with NDMP

Tape devices and configurations you can use with the filer

The filer can read from or write to these devices when using NDMP:

- Stand-alone tape drives or tapes within a tape library attached to the filer
- Tape drives or tape libraries attached to the workstation that runs the backup application
- Tape drives or tape libraries attached to a workstation or filer on your network
- NDMP-enabled tape libraries attached to your network

Restriction for tape libraries

If you use NDMP to back up the filer to attached tape libraries, be aware that if you want to use NDMP to manage your tape library, you must set the tape stacker autoload setting to Off. If the autoload setting is set to On, the filer uses the tape stacker the same way it uses a stand-alone tape drive and does not allow medium changer operations to be controlled by the NDMP backup application.

Naming conventions for tape libraries

Historically, the following names were always used to refer to tape libraries:

- mc*n* or /dev/mc*n*
- spt*n* or /dev/spt*n*

In a specific tape library name, *n* is a number. For example, mc0, spt0, /dev/mc0, and /dev/spt0 all refer to the same library.

Now, tape libraries can also be aliased to worldwide names (WWNs). For more information about tape aliasing, see “Assigning tape aliases” on page 303.

To see what tape libraries are recognized by your system, use the `sysconfig -m` command on the filer console. To see what names are currently assigned to any libraries, use the `storage show mc` command on the filer. Tape aliasing is also used to refer to tape drives, and you can see the aliases of tape drives using the `storage show tape` command.

Example

Below is an example of the output you might see from the `storage show mc` command:

```
filerA> storage show mc
Media Changer:          2.3
Description:            SPECTRA 10000
Serial Number:          7030290500
World Wide Name:        WWN[2:000:0090a5:00011c]
Alias Name(s):          mc0
Device State:           available (does not support reservations)
```

Preparing a filer for basic NDMP backup application management

About NDMP preparation

To enable a filer for basic management by a commercial NDMP backup application, you must enable the filer's NDMP support, specify the backup application's configured NDMP version, host IP address, and authentication method. If an operator with non-root privileges to the filer is using a backup application, a filer-generated NDMP-specific password must be used by that user to carry out backup operations on that filer.

To prepare a filer for NDMP management, complete the following steps.

Step	Action
1	Enable NDMP. At the console command line of target filer, enter: **`ndpmd on`** For details see "Enabling or disabling NDMP service" on page 433.
2	Specify the NPMD version to support on your filer. The version must match the version configured for your NDMP backup application. Enter: **`ndmpd version {2|3|4}`** For details see "Specifying a preferred network interface" on page 441.
3	If you want to specify a restricted set of NDMP backup application hosts that can connect to the filer, enter: **`options ndmpd.access`** ***`hosts`*** *hosts* is a comma-separated list of host names or IP addresses of nodes permitted to start NDMP sessions with the filer. If you do not wish to restrict NDMP access to this filer by host, do not set this option. By default NDMP does not restrict access by host. For details see "Specifying NDMP access by host or interface" on page 434.

Step	Action
4	Specify the authentication type (plaintext, challenge, or plaintext *and* challenge) required for an NDMP connection to this filer. For example: **`options ndmpd authtype plaintext,challenge`** This setting must include the authentication type supported by the NDMP backup application. For details see "Specifying the NDMP authentication type" on page 435.
5	If operators with non-root privilege on the filer will be carrying out tape backup operations through the NDMP backup application: ◆ Make sure they have a useradmin account on the filer. If they do not, enter: **`useradmin useradd username`** ◆ Display that user's system-generated NDMP-specific password. Enter: **`ndmpd password username`** When running the NDMP backup application, the user must input this NDMP-specific password to connect to this filer and carry out NDMP backup and restore operations. For details, see "Obtaining a generated NDMP-specific password for non-root administrators" on page 437.

Enabling or disabling NDMP service

What enabling NDMP service does

Enabling NDMP service on your filer allows NDMP-compliant data protection applications to communicate with the filer.

Enabling or disabling NDMP service

To enable or disable NDMP service, complete the following step.

Step	Action
1	Enter the following command: **`ndmpd {on|off}`** Use `on` to enable NDMP. Use `off` to disable NDMP. After you enter `ndmpd off`, the filer continues processing all requests on already established sessions, but rejects new sessions.

Note

The setting of the `ndmpd on|off` command is persistent across reboots.

Managing NDMP security features

About NDMP security features

Data ONTAP provides the following commands to enable you to manage and monitor the security of NDMP connections to the filer.

- The `options ndmpd.access` command enables you to restrict which hosts can run NDMP sessions with the filer. For details see "Specifying NDMP access by host or interface" on page 434.
- The `ndmpd authtype` command enables you to specify the authentication method (plaintext, challenge, or both) through which users are allowed to start NDMP sessions with the filer. For details see "Specifying the NDMP authentication type" on page 435.
- The `options ndmpd.connectlog` command allows you to enable or disable logging of NDMP connections attempts with the filer. For details see "Enabling or disabling NDMP connection logging" on page 436.
- The `ndmpd password` command enables even administrators with non-root level privilege on the filer to obtain a system-generated secure password. This is the password that will allow them to carry out NDMP operations, either through an NDMP-compliant backup application or through the filer command-line interface (CLI) that require a password. For details see "Obtaining a generated NDMP-specific password for non-root administrators" on page 437.

Specifying NDMP access by host or interface

You can use the `options ndmpd.access` configuration to specify the hosts or interfaces through which NDMP sessions are permitted. If you use this option to list specific authorized hosts or interfaces, NDMP sessions will be refused through any host or interface that is not listed by this option. Conversely, you can also specify hosts or interfaces to block from NDMP sessions.

To specify permitted hosts or interfaces, complete the following steps.

Step	Action
1	Start a console session on the filer to which you want to restrict NDMP access.

Step	Action
2	Enter the following command: **`options ndmpd.access {all|legacy|host[!]=`*`hosts`*`|if[!]=`*`interfaces`*`}`** host=*hosts* is a parameter string that allows a specified host or a comma-separated list of hosts to run NDMP sessions on this filer. The hosts can be specified by either hostname or IP address. host!=*hosts* is a parameter string that blocks a specified host or a comma-separated list of hosts from running NDMP sessions on this filer. The hosts can be specified by either hostname or IP address. if=*interfaces* is a parameter string that allows NDMP sessions through a specified interface or a comma-separated list of interfaces on this filer. if!=*interfaces* is a parameter string that blocks NDMP sessions through a specified interface or a comma-separated list of interfaces on this filer. `all` is the default value, which permits NDMP sessions with any host. `legacy` restores previous values in effect before a Data ONTAP version upgrade. In the case of Data ONTAP 6.2, the legacy value is equal to `all`.

Specifying the NDMP authentication type

Data ONTAP supports two methods, plaintext and challenge, for authenticating NDMP access to a filer.

You can use the `options ndmpd.authtype` command to specify whether a filer will accept plaintext, challenge, or both plaintext and challenge methods to authenticate NDMP session requests. To specify the authentication type, complete the following steps.

Step	Action
1	Start a console session on the filer whose NDMP authentication method you want to specify.

Step	Action

2	Enter *one* of the following command lines:
	`options ndmpd.authtype challenge`
	`options ndmpd.authtype plaintext`
	`options ndmpd.authtype plaintext,challenge`
	The first command line accepts the challenge authentication method, generally the preferred and more secure authentication method.
	The second command line accepts the plaintext authentication method, in which the login password is transmitted as clear text.
	The third command line accepts both challenge and plaintext authentication methods.
	If you are carrying out NDMP operations through a backup application, the authentication type or types you specify on this command line must include the types supported by that backup application.

Enabling or disabling NDMP connection logging

By default Data ONTAP logs NDMP connection attempts in the /etc/syslog file. These entries enable an administrator to determine whether and when authorized or unauthorized individuals are attempting to start NDMP sessions.

To enable or disable logging, complete the following step.

Step	Action
1	Start a console session on the filer on which you want to enable or disable NDMP connection monitoring.
2	Enter the following command: `options ndmpd.connectlog.enabled {on|off}` The value you set for this option will persist across filer reboots.

Step	Action
3	If you want to check attempted NDMP connection activity, use your UNIX or Windows Admin host to view your filer's /etc/syslog file. Entries recording attempted NDMP connections or operations will display the following fields: ◆ time ◆ thread ◆ NDMP request and action (allow or refuse) ◆ NDMP version ◆ session ID ◆ source IP (address of where the NDMP request originated) ◆ destination IP (address of the filer receiving the NDMP request) ◆ source port (port through which the NDMP request was transmitted) ◆ filer port through which the NDMP request was received **Example entry:** `Friday Sept 13:16:45:17GMT ndmpd.access allowed for version =4, sessid=34, from src ip = 172.29.19.40, dst ip =172.29.19.95, src port = 63793, dst port = 10000.`

Obtaining a generated NDMP-specific password for non-root administrators

Administrators who have an account on a filer, but do not have root status on that filer need to input a special NDMP-specific password when carrying out NDMP-related operations on the filer. This password is a system-generated string derived from the non-root administrator's regular filer account password.

The non-root administrator uses this NDMP-specific password for any NDMP backup and restore operation, in either a backup application or CLI environment, that requires password input.

The non-root administrator obtains an NDMP-specific password by completing the following steps:

Step	Action
1	Start a console session on the filer you want to access.

Step	Action

2	Enter the following command:
	`ndmpd password username`
	username is the username of the non-root administrator.
	The system returns a 16 character string, for example:
	`>ndmp password barbaraD`
	`>ndmp password: QM12N%$cnaFWPBVe`
	You use this password in any current or future NDMP operation that requires password input.
	Note
	This NDMP-specific password is valid until you change the password to your regular account.

3	If you change the password to your regular filer account, repeat this procedure to obtain your new system-generated NDMP-specific password.

Specifying the NDMP version

Why you need to specify the NDMP version

Data ONTAP provides full support for NDMP versions 2 to 4. Data ONTAP is shipped with the NDMP version set to 4, as both the default version and the maximum version.

The filer and the backup application must agree on a version of NDMP to be used for each NDMP session. When the backup application connects to the filer, the filer sends the default version back. The application can choose to use that default version and continue with the session. However, if the backup application uses an earlier version, it begins version negotiation, asking if each version is supported, to which the filer responds yes or no.

The `ndmp version` command sets an upper limit (maximum version) as well as a default version for the versions that the filer says, during the negotiation process, that it supports.

Because some backup applications do not support version negotiation, the `ndmp version` command controls the maximum and default NDMP version allowed. If you know your backup application does not support NDMP version 4 and does not negotiate versions, you can use this command to define the maximum version Data ONTAP supports so that the application can operate correctly.

Displaying the NDMP version

To see the maximum version of NDMP the filer is currently set to use, complete the following step.

Step	Action
1	Enter the following command: **`ndmpd version`** **Result:** The maximum version that NDMP currently allows to be used is displayed.

Specifying the NDMP version

To specify the NDMP version you want, complete the following step.

Step	Action
1	Enter the following command: **`ndmpd version n`** *n* is the number of the version you want to specify. Options available are 2, 3, and 4. The default maximum version is 4.

Note

The `ndmpd version` command is persistent across filer reboots.

Specifying a preferred network interface

Why to specify a preferred network interface

You can specify the preferred filer network interface to be used when establishing an NDMP data connection to another filer.

By default, an NDMP data connection uses the same network interface as the NDMP control connection established by the NDMP backup application. However, when a data connection between NDMP-enabled filers needs to be established over an alternate network, it is necessary to specify the filer's interface through which the alternate network will be accessed.

For example, a UNIX or NT resident NDMP backup application and multiple filers can be interconnected via a corporate network. The same filers can also be interconnected via an isolated private network. To minimize load on the corporate network, the `ndmpd.preferred_interface` option can be used to direct all NDMP data connections over the isolated private network.

How to specify a preferred network interface

To specify the preferred network interface to be used for NDMP data connections, complete the following step.

Step	Action
1	Enter the following command: **`options ndmpd.preferred_interface`** ***`interface`*** *interface* identifies the network interface to be used for all NDMP data connections. Any network interface providing TCP/IP access can be specified. If no argument is specified, the command returns the name of the interface currently configured for data connections. If no interface is currently set, it reports *disable*. You can find the available network interfaces by using the `ifconfig -a` command.

Note

The `ndmpd.preferred_interface` option is persistent across filer reboots.

How to turn off data connection specification

To disable a preferred network interface specification and force the NDMP default interface to be used for data connections, complete the following step.

Step	Action
1	Enter the following command: **`options ndmpd.preferred_interface disable`** **Result:** NDMP does not use the preferred interface for data connection. The default value for the `ndmp.preferred_interface` option is *disable*.

Displaying NDMP session status

Reasons to display NDMP session status

You can view either general status information or detailed session information about one or more NDMP sessions:

- View general status information to determine whether the session is operating as expected.
- View detailed session information to help you debug an NDMP session if there are problems.

How to display general status information about an NDMP session

To display general NDMP status information, complete the following step.

Step	Action
1	Enter the following command: **`ndmpd status [`*`session`*`]`** *session* is the specific session number you want the status of, from 0 to 99. Leave *session* blank to display the status of all current sessions.

Example

In the following example, the command displays information about session 4:

```
filerA> ndmpd status 4

ndmpd ON.
Session: 4
  Active
  version:                3
  Operating on behalf of primary host.
  tape device:    not open
  mover state:    Idle
  data state:     Idle
  data operation: None
```

How to display detailed NDMP session information

To display detailed NDMP session information, complete the following step.

Step	Action
1	Enter the following command: **ndmpd probe [*session*]** *session* is the number of the session you want to probe, from 0 to 99. Leave *session* blank to display detailed information about all sessions.

Example

In the following example, the command shows detailed status of session 4:

```
filerA> ndmpd probe 4

ndmpd ON.
Session: 4
  isActive:               TRUE
  protocol version:       3
  effHost:                Local
  authorized:             FALSE
  client addr:            10.10.10.12.47154
  spt.device_id:          none
  spt.ha:                 -1
  spt.scsi_id:            -1
  spt.scsi_lun:           -1
  tape.device:            rst0a
  tape.mode:              Read/Write
  mover.state:            Active
  mover.mode:             Read
  mover.pauseReason       N/A
  mover.haltReason        N/A
  mover.recordSize:       10240
  mover.recordNum:        315620
  mover.dataWritten:      3231948800
  mover.seekPosition:     0
  mover.bytesLeftToRead:  0
  mover.windowOffset:     0
  mover.windowLength:     -1
  mover.position:         0
  mover.connect.addr_type:LOCAL
  data.operation:         Backup
  data.state:             Active
```

```
data.haltReason:        N/A
data.connect.addr_type: LOCAL
data.bytesProcessed:    3231989760
```

Displaying NDMP debug messages

What debug messages are

NDMP debug messages provide a detailed description of all active NDMP sessions. The amount of information displayed by a debug message is determined by the ndmpd debug level specified by the user. By default, debug messages are disabled.

Where the filer saves debug messages

Debug messages can be output to the filer console, the NDMP session log file, or both. The `ndmpd debug` command is used to specify where debug messages will be output. By default, debug messages are output to both the filer console and the NDMP session log.

About the NDMP session log files

NDMP session log files are stored in the /etc/log directory. The file name is ndmplog.*yyyymmdd*, where *yyyy* is the year, *mm* is the month, and *dd* is the date. For example, the session log file generated on December 5, 2000, is named ndmplog.20001205. The session log file can contain information about one or more NDMP sessions. If multiple NDMP sessions take place on the same day, Data ONTAP saves the information about all sessions to the same session log file. Before generating a fresh NDMP session log file, Data ONTAP deletes all files more than eight days old. Data ONTAP keeps a maximum of eight session log files: one each for the past seven days plus the current day.

Enabling the NDMP debug log messages

To enable NDMP debug logging, complete the following step.

Step	Action
1	Enter the following command: **`ndmpd debug n`** *n* specifies the debug level from 0 to 70. The default level is 0. Use 0 for the level to turn the debug messages off. Use a nonzero value for the level to turn the debug messages on. The levels supported are: 10—Displays connections being made and connections being closed. 30—Displays information regarding the actual NDMP messages such as the message type, sequence numbers, and timestamps. This level also prints out the NDMP errors and some of the relevant fields of the NDMP message. 50—Same as level 30, but includes display of environment variables as well as any exceptions issued by the NDMP server implementation. 70—Same as level 50, but includes display of tape and SCSI command descriptor blocks (CDBs) sent. **Note** CDBs are used for low-level tape and medium changer control.

Displaying the NDMP debug log level

To see the NDMP debug levels currently set, complete the following step.

Step	Action
1	Enter the following command: **`ndmpd debug`** **Result:** The current NDMP debug level and toggles are displayed.

Example:

```
filerA> ndmpd debug
ndmpd debug verbose: 0
ndmpd debug stack trace: false
ndmpd debug screen trace: true
ndmpd debug file trace: true
```

Changing NDMP debug log messages

You use the `ndmp debug` command to choose whether to display the debug messages on the filer console, to write the debug messages into the NDMP debug log file, and to print out stack traces for any exceptions issued by the NDMP server implementation. To toggle these options, complete the following step:

Step	Action
1	Enter the following command: **`ndmpd debug stack|screen|trace`** `stack` toggles stack trace printouts on or off. `screen` toggles printouts to the filer console on or off. `trace` toggles printouts to the NDMP log file on or off.

Displaying an NDMP session log file

To display an NDMP session log file, complete the following step.

Step	Action	
1	**From a UNIX environment...**	**From an NT environment...**
	Mount the root volume of the filer to a UNIX client and view the contents of the NDMP session log file using the `cat` or `more` UNIX commands or a text editor.	Map the root volume of the filer to the NT system and view the contents of the NDMP session log file using WordPad, Notepad, or an equivalent text-viewing application.

Terminating an NDMP session

How to terminate an NDMP session

You terminate an NDMP session if the session is not responding.

To terminate an NDMP session, complete the following step.

Step	Action
1	Enter the following command: **`ndmpd kill session #`** *session #* is the specific NDMP session you want to terminate, from 0 to 99. If you want to terminate all NDMP sessions, enter the following command. **`ndmpd killall`** These commands allow nonresponding sessions to be cleared without the need for a reboot, since the `ndmpd off` command waits until all sessions are inactive before turning off the NDMP service.

Copying data with ndmpcopy

What ndmpcopy is

The `ndmpcopy` command enables a filer administrator to transfer file system data between filers that support NDMP v3 or v4 and the UNIX file system (UFS) dump format.

Using the `ndmpcopy` command, you can carry out both full and incremental data transfers; however, incremental transfers are limited to a maximum of two levels (one full and up to two incremental). You can transfer full or partial volumes, qtrees, or directories, but not individual files.

`ndmpcopy` functions as a simple NDMP data management application (backup application) that performs data transfers by initiating a backup operation on the source filer and recovery operation on the destination filer. It establishes separate NDMP control connections with the source and destination filers, creates a single NDMP data connection between the filers, and initiates backup and recovery operations that result in the desired data transfer between the source and destination.

You can invoke it at the command line of the source filer, the destination filer, or a filer that is neither the source nor the destination of the data transfer.

You can also invoke `ndmpcopy` on a single filer that is both the source and the destination of the data transfer. See the `ndmpcopy` examples, next, for more information on supported data transfer topologies.

How to copy data using ndmpcopy

To copy data within a filer or between filers using `ndmpcopy`, complete this step.

Note

The command can also be entered from a filer that is not the source or the destination.

Step	Action
1	Enter the following command: **`ndmpcopy [options][source_filer:]source_path [destination_filer:]destination_path`** *options* can be one or more of the following: `-sa` *username:*[*password*] source authorization specifies the user name and password for connecting to the source filer. **Note** If you specify a non-root user who requires a password, you must specify that user's system-generated NPMD-specific password rather than that user's regular filer account password. See "Enabling or disabling NDMP connection logging" on page 436. `-da` *username:*[*password*] destination authorization specifies the user name and password for connecting to the destination filer. `-st` {`challenge`|`text`} sets the source authentication type to be used when connecting to the source filer. `-dt` {`challenge`|`text`} sets the destination authentication type to be used when connecting to the destination filer. By default, challenge is the authentication type used. The text authentication type exchanges the user name and password in clear text. The challenge authentication type exchanges the user name and password in encrypted form. `-l` *level* sets the dump level used for the transfer to the specified value of *level.* Valid values for *level* are 0, 1, and 2, where 0 indicates a full transfer and 1 or 2 an incremental transfer. The default is 0. `-d` enables generation of `ndmpcopy` debug log messages. `ndmpcopy` debug log files appear in the root volume /etc/log directory. The `ndmpcopy` debug log file names are in the form ndmpcopy.*yyyymmdd*. `-f` enables forced mode. This mode enables overwriting system files in the /etc directory on the root volume. See Example 5 for details. `-h` prints the help message. *source_filer* and *destination_filer* can be host names or IP addresses

Examples of the ndmpcopy command

In these examples, network host names are used for the filers. ("myhost" is used for a local filer and "remotehost1" and "remotehost2" are used for remote filers.) If you specify host names when you use the `ndmpcopy` command, the filer running the `ndmpcopy` command should be able to resolve these names to their IP addresses. In UNIX, you can use the `ping` command to make sure that host name resolution works.

Example 1: This command migrates data from a source path (*source_path*) to a different destination path (*destination_path*) on the same filer (myhost).

```
myhost> ndmpcopy -sa username:password -da username:password
myhost:/vol/vol0/source_path myhost:/vol/vol0/destination_path
```

This shorter form of the command achieves the same purpose:

```
myhost> ndmpcopy /vol/vol0/source_path /vol/vol0/destination_path
```

Because you are running the `ndmpcopy` command on myhost and the source and destination filer are the same as myhost, you can omit the source and destination filer names on the `ndmpcopy` command line. When your `ndmpcopy` command is running on the same filer as the source filer or destination filer, you can also omit the `-sa` or `-da` options.

Example 2: This command migrates data from a source path (*source_path*) to a different destination path (*destination_path*) on remotehost1.

```
myhost> ndmpcopy -da username:password /vol/vol0/source_path
remotehost1:/vol/vol0/destination_path
```

The destination filer must be specified in this case, because it is a remote filer. The destination authorization is needed, but not the source authorization.

Example 3: This command migrates data from a source path (*source_path*) on remotehost2 to a destination path (*destination_path*) on myhost.

```
myhost> ndmpcopy -sa username:password -st text
remotehost2:/vol/vol0/source_path /vol/vol0/destination_path
```

The source authentication type specified by `-st` is text. The `ndmpcopy` command tool running on myhost will authenticate with the source filer using text authentication.

Example 4: This command migrates data from a source path (*source_path*) on remotehost1 to a destination path (*destination_path*) on remotehost2.

```
myhost> ndmpcopy -sa username:password -da username:password -l 1
remotehost1:/vol/vol0/source_path
remotehost2:/vol/vol0/destination_path
```

The `-l 1` option is used to do a level 1 transfer.

Example 5: Without the `-f` option, the /etc directory and its contents on the root volume of remotehost1 are protected from being overwritten with the /etc directory from myhost, preventing unintentional changing of the system characteristics after the root volume migration is completed.

```
myhost> ndmpcopy -da username:password /vol/rootvol
remotehost1:/vol/rootvol
```

To overwrite intentionally the /etc directory during the root volume migration, use the `-f` flag as in the following example.

```
myhost> ndmpcopy -da username:password -f /vol/rootvol
remotehost1:/vol/rootvol
```

Event Logging for Backup and Restore 12

About this chapter

This chapter describes how Data ONTAP automatically records the events that occur during dump and restore operations in log files. You can use these event logs to monitor, manage, and diagnose problems with your data backup and restore operations.

Topics in this chapter

This chapter discusses the following topics:

- "About event logging for backup and restore" on page 456
- "Event log messages" on page 457
- "Enabling or disabling event logging" on page 462

About event logging for backup and restore

What event logging is

Data ONTAP automatically logs significant events and the times at which they occur during dump and restore operations. All dump and restore events are recorded in a log file named backup in the /etc/log/ directory. Event logging is turned on by default.

Reasons for viewing event log files

You might want to view event log files for the following reasons:

- To find out whether a nightly backup was successful
- To gather statistics on backup operations
- To use information contained in past event log files to help diagnose problems with dump and restore operations

Log file rotation

Once every week, the log files are rotated. The /etc/log/backup file is copied to /etc/log/backup.0, the /etc/log/backup.0 file is copied to /etc/log/backup.1, and so on. The system saves the log files for up to six weeks; therefore, you can have up to seven message files (/etc/log/backup.0 through /etc/log/backup.5 and the current /etc/log/backup file).

Event log files in takeover mode

If a takeover occurs in a cluster, the set of backup log files for the takeover filer remains separate from the backup log files for the failed filer.

Event log messages

Event log message format

For each event, a message is written to the backup log file in the following format:

type timestamp identifier event (event_info)

Example: `dmp Thu Apr 5 18:54:56 PDT /vol/vol0/home(5) Start (level 0, NDMP)`

Type field

Each log message begins with one of the type indicators described in the following table.

Type	Meaning
`log`	Logging event
`dmp`	Dump event
`rst`	Restore event

Timestamp field

The timestamp field shows the date and time of the event.

Identifier field

The identifier field for a dump event includes the dump path and the unique ID for the dump. The identifier field for a restore event uses only the restore destination path name as a unique identifier. Logging-related event messages do not include an identifier field.

Start and stop logging events

The event field of a message that begins with `log` contains one of the events shown in the following table.

Event	Meaning
`Start_Logging`	Indicates the beginning of logging or that logging has been turned back on after being disabled.
`Stop_Logging`	Indicates that logging has been turned off.

Dump events

The event field for a dump event contains an event type followed by event-specific information in parentheses. The following table describes the events, their meanings, and the related event information that might be recorded for a dump operation.

Event	Meaning	Event Information
`Start`	The beginning of a dump or NDMP dump	Dump level and the type of dump
`Restart`	The beginning of a dump restart	Dump level
`End`	Dump completes successfully	Amount of data processed
`Abort`	The operation aborts	Amount of data processed
`Options`	Specified options are listed	All options and their associated values, including NDMP options
`Tape_open`	The tape is open for read/write	The new tape device name
`Tape_close`	The tape is closed for read/write	The tape device name
`Phase-change`	Dump is entering a new processing phase	The new phase name
`Error`	Dump encounters an unexpected event	Error message

Event	Meaning	Event Information
`Snapshot`	A snapshot is created or located	The name and time of the snapshot
`Base_dump`	A base dump entry in the etc/dumpdates files has been located	The level and time of the base dump (for incremental dumps only)

Example for dump

The following is an example of the output for a dump operation:

```
dmp Thu Sep 20 01:11:22 GMT /vol/vol0/(1) Start (Level 0)

dmp Thu Sep 20 01:11:22 GMT /vol/vol0/(1) Options (b=63, B=1000000,
u)

dmp Thu Sep 20 01:11:22 GMT /vol/vol0/(1) Snapshot
(snapshot_for_backup.6, Sep 20 01:11:21 GMT)

dmp Sep 20 01:11:22 GMT /vol/vol0/(1) Tape_open (nrst0a)

dmp Sep 20 01:11:22 GMT /vol/vol0/(1) Phase_change (I)

dmp Sep 20 01:11:24 GMT /vol/vol0/(1) Phase_change (II)

dmp Sep 20 01:11:24 GMT /vol/vol0/(1) Phase_change (III)

dmp Sep 20 01:11:26 GMT /vol/vol0/(1) Phase_change (IV)

dmp Sep 20 01:14:19 GMT /vol/vol0/(1) Tape_close (nrst0a)

dmp Sep 20 01:14:20 GMT /vol/vol0/(1) Tape_open (nrst0a)

dmp Sep 20 01:14:54 GMT /vol/vol0/(1) Phase_change (V)

dmp Sep 20 01:14:54 GMT /vol/vol0/(1) Tape_close (nrst0a)

dmp Sep 20 01:14:54 GMT /vol/vol0/(1) End (1224 MB)
```

There are five phases in a dump operation (map files, map directories, dump directories, dump files, and dump ACLs).

The log file for a dump operation begins with either a `Start` or `Restart` event and ends with either an `End` or `Abort` event.

Restore events

The event field for a restore event contains an event type followed by event-specific information in parentheses. The following table describes the events, their meanings, and the related event information that might be recorded for a restore operation.

Event	Meaning	Event Information
`Start`	The beginning of a restore or NDMP restore	Restore level and the type of restore
`Restart`	The beginning of a restore restart	Restore level
`End`	Restore completes successfully.	Number of files and amount of data processed
`Abort`	The operation aborts.	Number of files and amount of data processed
`Options`	Specified options are listed.	All options and their associated values, including NDMP options
`Tape_open`	The tape is open for read/write.	The new tape device name
`Tape_close`	The tape is closed for read/write.	The tape device name
`Phase-change`	Restore is entering a new processing phase.	The new phase name
`Error`	Restore encounters an unexpected event.	Error message

Example for restore

The following is an example of the output for a restore operation:

```
rst Thu Sep 20 02:24:22 GMT /vol/rst_vol/ Start (level 0)

rst Thu Sep 20 02:24:22 GMT /vol/rst_vol/ Options (r)

rst Thu Sep 20 02:24:22 GMT /vol/rst_vol/ Tape_open (nrst0a)

rst Thu Sep 20 02:24:23 GMT /vol/rst_vol/ Phase_change (Dirs)
```

```
rst Thu Sep 20 02:24:24 GMT /vol/rst_vol/ Phase_change (Files)

rst Thu Sep 20 02:39:33 GMT /vol/rst_vol/ Tape_close (nrst0a)

rst Thu Sep 20 02:39:33 GMT /vol/rst_vol/ Tape_open (nrst0a)

rst Thu Sep 20 02:44:22 GMT /vol/rst_vol/ Tape_close (nrst0a)

rst Thu Sep 20 02:44:22 GMT /vol/rst_vol/ End (3516 files, 1224 MB)
```

There are two phases in a restore operation (restore directories and restore files).

The log file for a restore operation begins with either a `Start` or `Restart` event and ends with either an `End` or `Abort` event.

Example of an aborted restore

The following is an example of the output of an aborted restore operation:

```
rst Thu Sep 20 02:13:54 GMT /rst_vol/ Start (Level 0)

rst Thu Sep 20 02:13:54 GMT /rst_vol/ Options (r)

rst Thu Sep 20 02:13:54 GMT /rst_vol/ Tape_open (nrst0a)

rst Thu Sep 20 02:13:55 GMT /rst_vol/ Phase_change (Dirs)

rst Thu Sep 20 02:13:56 GMT /rst_vol/ Phase_change (Files)

rst Thu Sep 20 02:23:40 GMT /vol/rst_vol/ Error (Interrupted)

rst Thu Sep 20 02:23:40 GMT /vol/rst_vol/ Tape_close (nrst0a)

rst Thu Sep 20 02:23:40 GMT /vol/rst_vol/ Abort (3516 files, 598
MB)
```

Enabling or disabling event logging

Turning event logging on or off

To turn event logging on or off, complete the following step.

Step	Action
1	Enter the following command to enable or disable event logging: **`options backup.log.enable [on | off]`** *on*—turns event logging on. *off*—turns event logging off. **Note** Event logging is turned on by default.

Database Protection Using NVFAIL 13

About this chapter

This chapter describes how Data ONTAP provides database protection using the `nvfail` option.

Topics in this chapter

This chapter discusses the following topics:

- "Understanding the nvfail option" on page 464
- "Enabling and disabling database file protection" on page 466
- "Using the nvfail_rename file for additional database protection" on page 467

Understanding the nvfail option

How Data ONTAP provides database file protection

Data ONTAP provides database file protection through the `nvfail` option of the `vol options` command. The `nvfail` option enables Data ONTAP to detect nonvolatile RAM (NVRAM) inconsistencies at boot time or while taking over in a cluster failover configuration.You use it to warn database administrators of NVRAM problems that can compromise database validity. If Data ONTAP finds any problems, database instances stop responding or shut down, and Data ONTAP sends error messages to the console to alert you to check the state of the database.

How to provide additional protection for database files

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

How nvfail works

When you enable `nvfail`, one of the following processes takes place during bootup.

If Data ONTAP...	Then...
Detects no NVRAM errors	File service starts normally.

If Data ONTAP...	Then...
Detects NVRAM errors and you use the optional nvfail_rename file	**1.** Data ONTAP returns a stale file handle (ESTALE) error to NFS clients trying to access the database, causing the application to stop responding, crash, or shut down. Data ONTAP then sends an error message to the filer console and log file. **2.** Data ONTAP renames database files specified in the nvfail_rename file by appending .nvfail to the original file names, making those files unavailable to both CIFS and NFS clients.
Detects NVRAM errors and you do not use the optional nvfail_rename file	**1.** Data ONTAP returns a stale file handle (ESTALE) error to NFS clients trying to access the database, causing the application to stop responding, crash, or shut down. Data ONTAP then sends an error message to the filer console and log file. **2.** No database files are renamed. When the application restarts, files are available to CIFS clients, even if you have not verified that they are valid. For NFS clients, files remain inaccessible as long as the file system is not remounted, independent of the application restarting.

Where to look for database file verification instructions

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

Enabling and disabling database file protection

Turning database protection on and off

To enable or disable the `nvfail` option, complete the following step.

Step	Action
1	Enter the following command: **`vol options volume_name nvfail [on|off]`** *volume_name* is the name of the volume. Use On to enable or Off to disable protection. The default setting is Off.

Using the nvfail_rename file for additional database protection

About the nvfail_rename file

This procedure enables you to use the optional nvfail_rename file when you want to rename database files after Data ONTAP detects NVRAM errors. This enables you to examine the files for consistency before clients can access them.

Restrictions

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

Creating the nvfail_rename file

To create the nvfail_rename file, complete the following steps.

Step	Action
1	Use an editor to create or modify the nvfail_rename file in the filer's /etc directory.
2	List the path name and file name of database files you want to protect, one file per line, within the nvfail_rename file. **Example:** `/vol/vol1/home/dbs/oracle-WG73.dbf`
3	Save the file.

Virus Protection for CIFS 14

About this chapter

This chapter discusses the virus scan feature for CIFS and how to configure it to scan files on a NetApp filer.

Topics in this chapter

This chapter discusses the following topics:

- "Understanding CIFS virus protection" on page 470
- "Setting up and starting virus scanning" on page 472
- "Specifying file types to be scanned" on page 475
- "Excluding file types to be scanned" on page 477
- "Specifying shares for scanning" on page 481
- "Displaying the scanner list" on page 487
- "Checking vscan information" on page 488
- "Setting and resetting the virus scan request timeout" on page 490
- "Allowing file access when the scan cannot be performed" on page 491
- "Controlling vfiler usage of host filer virus scanners" on page 492
- "Checking the status of virus-scanning options" on page 493
- "Stopping a virus scanner session" on page 494
- "Resetting the scanned files cache" on page 495
- "Enabling virus scan messages to CIFS clients" on page 496

Understanding CIFS virus protection

What CIFS virus protection is

CIFS virus protection is a Data ONTAP feature that allows a virus-scanning PC client running Network Appliance-compliant antivirus applications to provide on-access virus scanning of files on a filer. On-access virus scanning means that a file is scanned before a CIFS client is allowed to open it.

How CIFS virus scanning works

CIFS virus-scanning is carried out on dedicated PC clients running the Network Appliance-compliant antivirus application of your choice. When you enable the virus-scanning process through Data ONTAP on the filer, the virus-scanning application tells the filer to send file scanning requests.

The virus-scanning application watches for requests from the filer. Whenever the types of files you specify are opened or changed on the filer, Data ONTAP sends the PC client a request to scan the file.

The Data ONTAP virus-scanning process can scan multiple filers from a single PC client if your virus-scanning application performs this function. For more information about whether a specific virus-scanning application can accommodate scanning multiple filers, contact the manufacturer of your virus-scanning application.

Obtaining Network Appliance-compliant anti-virus software

In order to successfully scan for viruses on the filer, the anti-virus application installed on your PC client must be a version customized for Network Appliance filers and supplied by one of Network Appliance's anti-virus software partners.

For a listing of the major software vendors who supply anti-virus software compliant with Network Appliance, see the *Partner Solutions Catalog, Data Protection* at the Network Appliance web site, http://www.netapp.com.

You can also see the *Antivirus Scanning Best Practices Guide* in the Windows section of the Network Appliance Technical Library at http://www.netapp.com.

Default file types that are scanned

By default, files with the extensions shown in the following table are scanned when CIFS virus scanning is enabled.

??_	DL?	IM?	OFT	SMM
ARJ	DOC	INI	OLE	SWF
ASP	DOT	JS?	OV?	SYS
BAT	DRV	LZH	PIF	VBS
BIN	EML	MD?	POT	VS?
CAB	EXE	MPP	PP?	VXD
CDR	GMS	MPT	RAR	WBK
CL?	GZ?	MSG	RTF	WPD
COM	HLP	MSO	SCR	XL?
CSC	HT?	OCX	SHS	XML

Note

In this table, the "?" character is a wildcard that matches any character or no character. For example,"C??" matches "C", "CL", "CPP", "C++", and so on.

Caution

Using "???" will cause all files to be scanned, which may severely degrade performance, and is not recommended.

You can also configure Data ONTAP so that file extensions not in the default list are scanned, or only a subset of the default file extensions is scanned. For more information, see "Specifying file types to be scanned" on page 475.

Setting up and starting virus scanning

About virus-scanning clients

The commercial virus-scanning software compliant with Data ONTAP runs on selected PC clients with "Backup Operator" or higher user privilege connections to the target filers.

During installation of the commercial virus scan software on the selected PCs, you indicate the Data ONTAP filers to scan for viruses. Once the relationship is set up, the PCs running the virus scan software act as special *virus-scanning clients* that will virus scan files that are being opened by *other* CIFS clients and, if necessary, block the file operations and send a virus detection warning message to the requesting CIFS client.

Specifying the active virus-scanning clients

To set up PC clients as virus-scanning clients, complete the following steps.

Step	Action
1	If you have not already done so, obtain Network Appliance-compliant virus scan software described in "Obtaining Network Appliance-compliant anti-virus software" on page 470.
2	Make sure that the operators of the PC clients that you want to configure as virus-scanning clients are configured as "Backup Operators" or higher on the filers on which they will conduct virus scanning.
3	Install the NetApp-customized commercial virus scan software on the PCs that you want to configure as virus-scanning clients. Follow the directions that accompany the virus scan software product.
4	After installation and configuration of the virus scan software on the PCs is complete, confirm the success of the installation by listing the IP addresses of the PCs now configured as virus-scanning clients. At the filer console enter the following command: `vscan scanners` **Result:** The system displays a table listing the IP addresses of the active virus-scanning clients for this filer.

Step	Action
5	Leave the virus-scanning client on and connected to the filer or filers on which it is carrying out its virus scan operations.

Enabling and disabling virus scanning

To enable and disable virus scanning, complete the following step.

Step	Action
1	Enter the following command: **`vscan on [-f][on|off]`** `-f` forces virus scanning to be enabled even if no virus-scanning clients are available to scan files.

Designating secondary scanning clients

If you have more than one virus-scanning client configured to scan files on a filer, you can use the `vscan scanners secondary_scanners` command line to designate one or more of the virus-scanning clients as standby virus scanners. Standby virus scanners do not actively scan for viruses unless the other virus-scanning clients go down. At that point the secondary scanners take over the virus-scanning operation.

To set up secondary virus scanning, complete the following steps.

Step	Action

1	List the virus-scanning clients configured to scan on this filer. In the filer console enter the following command:
	`vscan scanners`
	Example:
	`>vscan scanners`
	`Virus scanners(IP and Name)  P/S ...`
	`----------`
	`132.132.59.12  \\XLAB-WTS  Pri ....`

Step	Action
2	Specify, by IP address, the PC clients you want to serve as standby virus scanners by entering the following command: **`vscan scanners secondary_scanners`** ***`scanner_ip`*`[,`*`scanner_ip`*`...]`** *scanner_ip* can be either of the following: ◆ IP addresses of one or more of the configured virus-scanning clients displayed in Step 1 ◆ IP addresses of PCs not yet configured as a virus-scanning client. **Note** If the IP address you entered belongs to a PC not yet configured as a virus-scanning client for this filer, you must configure it for this setting to take effect. **Example:** `vscan scanners secondary_scanners 132.132.59.14`

3	Use the `vscan scanners` command to confirm your configuration.
	Example:
	`>vscan scanners`
	`Virus scanners(IP and Name)  P/S ...`
	`----------`
	`132.132.59.14  \\BORIS-PC  Sec ...`
	`132.132.59.12  \\XLAB-WTS  Pri ....`
	
	`Secondary scanners IP address list`
	`132.132.59.14,10.20.30.40`
	Note
	In this example, the address 10.20.30.40 is the address of a PC that is enabled as a standby scanner but is not turned on; therefore it is not listed in the Virus scanners (IP and Name) table, but it is listed in the Secondary scanners IP address list.

Specifying file types to be scanned

About specifying file types

Use the `vscan extensions include` command line to list, specify, add to, or remove from the set of file types to be scanned on the filer.

Displaying files to be scanned

To display the list of extensions of file types to be scanned on the filer, complete the following step.

Step	Action
1	Enter the following command: **`vscan extensions include`** **Result:** The current list of extensions to be scanned is displayed.

Adding file types to be scanned

A default list of file extensions is made available when you enable virus scanning; however, you can specify additional file extensions not in the default list.

To add the extensions of file types to be scanned, complete the following step.

Step	Action
1	Enter the following command: **`vscan extensions include add`** ***`ext`*****`[,`*****`ext`*****`...]`** *ext* is the extension you want to add.

Example: `vscan extensions include add txt`

Note

Up to 255 file extensions can exist in the file extensions list.

Replacing the file types in the extensions list

You can replace the current list of file extensions with a new list of extensions that you choose.

To replace the `vscan` extensions list with a new list, complete the following step.

Step	Action
1	Enter the following command: **`vscan extensions include set ext[,ext...]`** *ext* is the extension you want to set.

Removing file types from the extensions list

To remove one or more file types from the extensions list, complete the following step.

Step	Action
1	Enter the following command: **`vscan extensions include remove ext[,ext...]`** *ext* is the extension you want to remove.

Resetting the file extensions list to the default

To reset the file extensions list to the default, complete the following step.

Step	Action
1	Enter the following command: **`vscan extensions include reset`** **Result:** The list of file extensions is set to the default list. See "Default file types that are scanned" on page 471.

Excluding file types to be scanned

About excluding file types

Some administrators, because of the proliferation of new file types (with new file name extensions) that might be stored on the filer, prefer to specify what file types to exclude, rather than include, in a virus scan. The practice of specifying files to exclude, lets you forgo scanning file types assumed to be safe, at the same time ensuring that any new, unfamiliar file type stored on the filer does get virus scanned.

Use the `vscan extensions exclude` command line to list, specify, add to, or remove from the set of file types to be excluded from virus scan on the filer.

Typical vscan extensions exclude usage

Typically, you will use the `vscan extensions exclude` command in combination with the `vscan extensions include` command. You use `include` to specify a general virus scan, and use `exclude` to specify the file types to be excepted from the virus scan.

To exclude file types from virus scanning, complete the following steps.

Step	Action
1	At the filer console enter the following command line specifying the extensions of all file types that you want to exclude from virus scan: **`vscan extensions exclude set ext[,ext...]`**
2	Enter the following command line to specify virus scan of all other file types: **`vscan extensions include set ???`** **Result:** This command line instructs the virus scan program to scan all file types stored on the filer. But the result of both the "exclude" and the "include" command lines together is that all file types are scanned except for the file types whose extensions have been specified in the `vscan extensions exclude` command line described in Step 1.

Displaying files to exclude from a scan

To display the list of extensions of file types to be excluded from a virus scan on the filer, complete the following step.

Step	Action
1	Enter the following command: **`vscan extensions exclude`** **Result:** The current list of extensions to be excluded from virus scan is displayed.

Specifying all the file types to exclude from a scan

To specify the list of extensions of file types that you want to exclude from virus scan, complete the following step.

Step	Action
1	Enter the following command: **`vscan extensions exclude set`** ***`ext`*****`[,`*****`ext`*****`...]`** *ext* is the extension or extensions that you want to set for the list of file types excluded from virus scan. **Note** Using the `set` parameter will replace completely any existing file type extensions in the exclude list with the extensions you specified in this command. If an exclude list already exists and you merely want to add to it, use the `vscan extensions exclude add` command.

Adding file types to exclude from a scan

To add the extensions of file types to be excluded from a virus scan, complete the following step.

Step	Action
1	Enter the following command: **`vscan extensions exclude add ext[,ext...]`** *ext* is the extension you want to add to the list of file types excluded from virus scan.

Example: `vscan extensions exclude add txt`

Note
Up to 255 file extensions can exist in the file extensions list.

Removing file types to exclude from a scan

To remove one or more file types to exclude from virus scan, complete the following step.

Step	Action
1	Enter the following command: **`vscan extensions exclude remove ext[,ext...]`** *ext* is the extension you want to remove from the list of file types excluded from virus scan.

Resetting the exclude file extensions to empty

To reset the file extensions list to the default, which is empty, complete the following step.

Step	Action
1	Enter the following command: **`vscan extensions exclude reset`** **Result:** The list of file extensions is set to the default empty value.

Exclude over include

If an extension for a file type is unintentionally entered in both the vscan include list (through the `vscan extensions include` command) and the vscan exclude list (through the `vscan extensions exclude` command, that file type will be excluded from the virus scan.

Specifying shares for scanning

About specifying shares for scanning

You can turn scanning on or off for shares you specify, either for any access or for read-only access. You can also create shares that you specify are or are not to be scanned, upon either any access or read-only access.

For detailed information

The following sections discuss the ways to turn scanning on and off for shares you specify or create:

- "Turning scanning on or off for shares" on page 482
- "Turning scanning on or off for read-only access for shares" on page 484
- "Adding shares with virus scanning turned on or off" on page 485

Turning scanning on or off for shares

Why you turn virus scanning off for a specified share

You might want to turn virus scanning off for files in a share if the users are restricted to trusted users, the files are restricted to read-only mode, or speed of access is more important than safety. The default state of a share is for virus scanning to be turned on.

How to turn virus scanning off for shares

To turn virus scanning off for files in a specified share, complete the following step.

Step	Action
1	Enter the following command: **`cifs shares -change share_name -novscan`** *share_name* is the name of the share for which you want to turn off virus scanning. **Result:** The virus-scanning application does not perform a virus scan when clients open files on this share. The setting is persistent after reboot.

How to turn virus scanning on for shares

To turn virus scanning on for files in a specified share, complete the following step.

Step	Action
1	Enter the following command: **`cifs shares -change share_name -vscan`** *share_name* is the name of the share for which you want to turn on virus scanning. **Result:** The virus-scanning application does a virus scan when clients open files on this share. The setting is persistent after reboot.

Note

You can set these share attributes for all CIFS user home directories by using `cifs.homedir` as the share name, for example, `cifs shares -change cifs.homedir -novscan`.

Turning scanning on or off for read-only access for shares

Why you turn virus scanning off for read-only access for shares

You might want to turn virus scanning off in a share for users who are opening files for read-only access to increase the speed of accessing the files. The default state of a share is for virus scanning to be turned on.

How to turn virus scanning off for shares

To turn virus scanning off for read-only access to files in a specified share, complete the following step.

Step	Action
1	Enter the following command: **`cifs shares -change share_name -novscanread`** *share_name* is the name of the share for which you want to turn off virus scanning. **Result:** The virus-scanning application does not perform a virus scan when clients open files on this share for read access. The setting is persistent after reboot.

How to turn virus scanning on for shares

To turn virus scanning on for read-only access for files in a specified share, complete the following step.

Step	Action
1	Enter the following command: **`cifs shares -change share_name -vscanread`** *share_name* is the name of the share for which you want to turn on virus scanning. **Result:** The virus-scanning application does a virus scan when clients open files on this share for read access. The setting is persistent after reboot.

Adding shares with virus scanning turned on or off

Why you add shares with virus scanning turned off

You might want to create a share with virus scanning turned off. The default state of a share is for virus scanning to be turned on.

How to add a share with virus scanning turned off

To add a share that has virus scanning turned off, complete the following step.

Step	Action
1	Enter the following command: **`cifs shares -add share_name /path -novscan`** *share_name* is the name of the share with virus scanning turned off that you want to create. *path* specifies where you want the share created. **Result:** Data ONTAP creates a share with virus scanning turned off.

How to add a share with virus scanning turned off for read-only access

To create a share with virus scanning turned off for read-only access, complete the following step.

Step	Action
1	Enter the following command: **`cifs shares -add share_name /path -novscanread`** *share_name* is the name of the share with virus scanning turned off for read-only access that you want to create. *path* specifies where you want the share created. **Result:** Data ONTAP creates a share with virus scanning turned off for read-only access.

Note

For backup purposes, you can create two shares on the same directory: one share with scanning disabled and a share-level ACL that allows access only to a backup account; the other share available to normal users and with scanning enabled. Backup can be performed on the share with no virus scanning and improved performance, while normal users continue to access the data through the regular share and get virus protection.

Displaying the scanner list

Displaying the scanner PC list

The scanner list identifies all scanner clients that are currently performing virus scanning for a filer. To display the scanner list, complete the following step.

Step	Action
1	Enter the following command: **`vscan scanners`** **Result:** The filer outputs the names of and information about the virus-scanning clients in a display like the one shown in the following paragraph.

```
Virus scanners(IP and Name)   Connect time (dd:hh:mm)  Reqs  Fails
-------------------------------------------------------------------
10.61.155.118  \\WIN2K-NAN    00:02:23                 2     0
10.60.129.152  \\WIN2K-RTB    00:00:01                 0     0
```

The *Connect time* field displays the length of time the scanner has been connected to the filer

The *Reqs* field displays the total number of requests sent by the filer to that scanner.

The *Fails* field displays the number of requests that failed.

Checking vscan information

How to check current vscan settings and information

You can check the on/off status of vscan, the current list of file extensions, the names of and information about the virus-scanning clients, the list of extensions to scan, the files scanned, and the number of scan failures on a filer.

To check the status of the vscan appliance, complete the following step.

Step	Action
1	Enter the following command: **vscan** **Result:** The filer outputs the on/off status of vscan, information about the virus-scanning clients, the list of extensions to scan, and the number of files scanned and number of scan failures in a display like the one shown in the following paragraph.

```
filerA> vscan

Virus scanning is enabled.

Virus scanners(IP and Name)  Connect time (dd:hh:mm)  Reqs    Fails
---------------------------------------------------------------------
10.61.155.118  \\WIN2K-NAN    00:02:23                 2     0
10.60.129.152  \\WIN2K-RTB    00:00:01                 0     0

List of extensions to scan:
??_,ASP,BAT,CDR,COM,CSC,DL?,DOC,DOT,EXE,GMS,GZ?,HLP,HT?,IM?,INI,JS
?,MD?,MPP,MPT,MSG,MSO,OCX,OLE,OV?,POT,PP?,RTF,SCR,SHS,SMM,SYS,VBS,
VS?,VXD,WBK,WPD,XL?,XML

List of extensions not to scan:

Number of files scanned: 28
Number of scan failures: 0
```

Note

The number of scan requests and failures shown in the table at the beginning of the output are not the same as the number of files scanned and scan failures shown at the end of the output. The first set of numbers reflects the number of requests and failures for this connection session; the second set of numbers reflects the total scans and failures since vscan was turned on.

Setting and resetting the virus scan request timeout

About the timeout setting

The virus scan timeout setting determines how long Data ONTAP will wait for a scanner to complete file scanning before requesting the status of the scan. The request will be repeated as often as necessary until the scan is complete or the host gives up. By default, the virus scan timeout is 30 seconds.

Setting the virus scan request timeout

To set the virus scan request timeout to something other than the default, complete the following step.

Step	Action
1	Enter the following command: **`vscan options timeout set value`** *value* is a setting from 1 to 45 seconds.

Resetting the virus scan request timeout

To reset the virus scan timeout back to the default, complete the following step.

Step	Action
1	Enter the following command: **`vscan options timeout reset`**

Allowing file access when the scan cannot be performed

About the mandatory_scan option

You can specify whether files can be accessed if the scan cannot be performed by using the `mandatory_scan` option of the `vscan` command. The default setting is On. When this option is set to On, file access is denied if a successful scan cannot be performed. When this option is set to Off, access is allowed even if no successful scan can be performed.

A scan might not be able to be performed if no scanner is available, if the request times out, or for other reasons.

Enabling and disabling mandatory scanning

To enable and disable mandatory file scanning, complete the following step.

Step	Action
1	Enter the following command: **`vscan options mandatory_scan [on|off]`**

Controlling vfiler usage of host filer virus scanners

What a vfiler is

The vfiler™ feature of the MultiStore™ software product is created by partitioning the storage and networking resources of a single filer so that it appears as multiple filers on the network. Each "filer" created as a result of the partitioning is called a vfiler. A vfiler, using the resources assigned, delivers file services to its clients as a filer does. For more information about vfilers, see the *Data ONTAP MultiStore Administration Guide.*

About controlling virus scanner usage

By default, vfilers can scan files using the virus scanners that are connected to the host filer. If you have several vfilers, each owned by a separate organization, you might want to disable a vfiler's ability to use the host filer's scanners if you have security concerns.

You can control whether vfilers are allowed to use the host filer's virus scanners by using the `use_host_scanners` option of the `vscan` command.

Enabling and disabling host scanner usage for vfilers

To enable or disable a vfiler's ability to use the virus-scanning clients recognized by the host filer, complete the following step.

Step	Action
1	From the vfiler console, enter the following command: **`vscan options use_host_scanners [on|off]`**

Note

This command does not run on the default vfiler.

Checking the status of virus-scanning options

Why to check the status of virus-scanning options

You check the status of the virus-scanning options to determine whether the options you want to use are set to the values you want.

How to check the status of virus-scanning options

To check the status of virus-scanning options, complete the following step.

Step	Action

1	Enter the following command:
	`vscan options`
	Result: The filer outputs the state of the virus-scanning options in a display like the following:
	`vscan options mandatory_scan     on`
	`vscan options timeout:           12 sec`
	`vscan options use_host_scanners  on`

Note

The `use_host_scanners` option applies only to vfilers and is displayed only if the `vscan options` command was run on a vfiler.

Stopping a virus scanner session

Reason for stopping a virus scanner session

At some time you might want to stop a virus scanner session. For example, you might want to stop a scanner session when you have to terminate CIFS on a filer or upgrade your antivirus program to a new version.

Stopping a specific scanner session

To stop a specific virus scanner session, complete the following step.

Step	Action
1	Enter the following command: **`vscan scanners stop`** ***`scanner_IP`*** *scanner_IP* is the IP address of the virus scanner you want to stop.

Resetting the scanned files cache

When to reset the scanned files cache

Data ONTAP caches information about previously scanned files to avoid rescanning those files. When you have a new virus-scanning signature file, you might want to reset the cache to rescan files that were scanned using an old signature file.

Resetting the cache

To reset the scanned files cache on a filer, complete the following step.

Step	Action
1	Enter the following command: `vscan reset`

Enabling virus scan messages to CIFS clients

Virus scan messages

You can enable Data ONTAP to return an explicit virus warning messages to clients if the virus-scanning application running on the filer detects that a file that CIFS client is trying to access is infected with a virus.

If this feature is not enabled, CIFS clients attempting to access virus-infected files that have been detected by virus scanning will simply receive a general “file unavailable” message.

Enabling the virus scan message

To enable the display of explicit virus warning messages to CIFS clients, complete the following step.

Step	Action
1	Enter the following command: **`vscan options client_msgbox {on|off}`** `on` enables the display of virus warning messages. `off` disables the display of virus warning messages.

Disaster Protection Using MetroCluster A

About this appendix

This appendix describes how to use the optional MetroCluster feature of Data ONTAP to protect data that requires continuous backup from loss and unavailability due to disasters such as prolonged power outages or natural disasters.

Note

This appendix describes protecting data on Network Appliance filers. For information about protecting data on other networked equipment, application servers, for example, see the documentation for that equipment.

Topics in this appendix

This appendix discusses the following topics:

- "Configuring disaster recovery using MetroCluster" on page 498
- "Recognizing a disaster" on page 503
- "Recovering from a disaster" on page 505

Configuring disaster recovery using MetroCluster

Definition of terms

This appendix uses the following terms when describing disaster recovery using filers in a MetroCluster configuration:

- Cluster—Two filers that are connected in a cluster configuration.
- Cluster node—One of the filers in the cluster.
- Local cluster node—The cluster node in the same area as the system administrator.
- Remote cluster node—The cluster node in the area affected by the disaster.
- Remote storage—The storage that is accessible to the local cluster node, but is located at the remote cluster node site.
- Site—The area in which a cluster node is located, for example, a building.

Mirrored volume configuration

The MetroCluster configuration employs SyncMirror and a cluster configured for mirrored volumes to build a system that provides data even after complete loss of one of the cluster nodes and the storage at that site. Mirrored volumes are configured as follows:

- The root volume must be mirrored with one plex at each site.
- Data volumes should be mirrored with one plex at each site.
- Plexes local to the filer at each site must be connected to pool0. Pool0 consists of the storage attached to host adapters in slots 3 through 7. See the *System Configuration Guide* at now.netapp.com and the *F800 Series Cluster Guide* for more information about filer slot assignments.

Note

The MetroCluster configuration can preserve data only if volumes are mirrored. Unmirrored volumes are not accessible if a disaster occurs; data can be lost.

MetroCluster configurations

There are two MetroCluster configurations. One configuration directly connects the two cluster nodes to each other. The other configuration uses Brocade switches to connect the two cluster nodes. The configuration you use depends on the distance between the two cluster nodes. The following sections discuss the two MetroCluster configurations:

- "MetroCluster configuration up to 500 meters" on page 499
- "MetroCluster configuration for distances up to 10 kilometers" on page 501

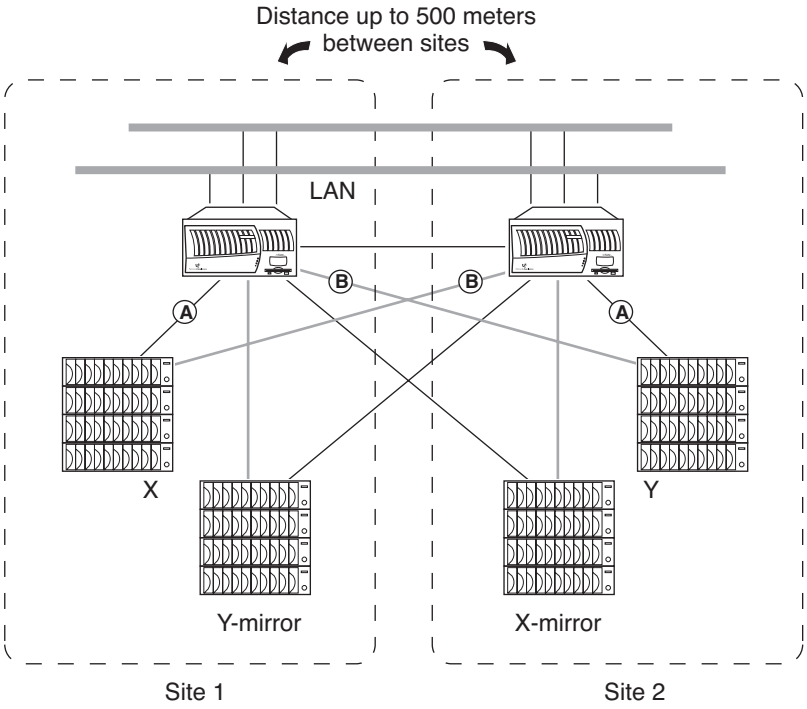
MetroCluster configuration up to 500 meters

Configuration overview

The cluster configuration for distances of up to 500 meters is the standard cluster-SyncMirror configuration using 500-meter interconnect cables between the two cluster nodes.

See the *F800 Series Cluster Guide* and the *Data Protection Guide* for detailed information about configuring clusters and using SyncMirror to mirror volumes, respectively.

The following figure illustrates the MetroCluster configuration.



Configuration requirements for distances under 500 meters

The MetroCluster configurations require the following:

- F800 series or FAS 900 series cluster pair configured for mirrored volume use
- The following licenses:
  - Cluster—Enables the two nodes to communicate and take over each other's functions in the event of a failure.
  - Syncmirror_local—Enables mirrored volumes.
  - Cluster_remote—Enables clustering and mirroring volumes over long distances.

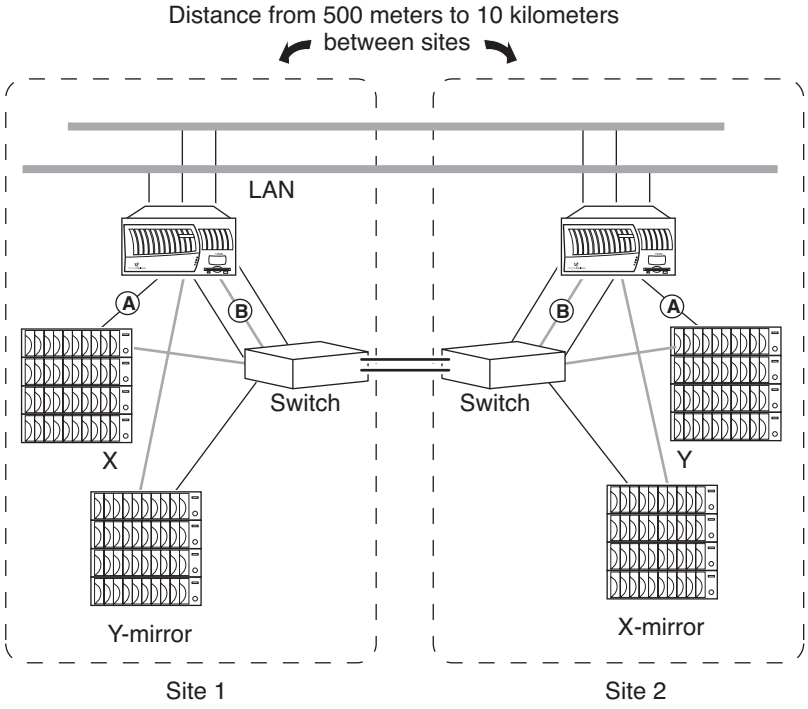
MetroCluster configuration for distances up to 10 kilometers

Configuration overview

The cluster configuration for distances between 500 meters and 10 kilometers uses a pair of Brocade 3800 switches to connect the two cluster nodes.

See the installation and configuration information in Appendix A and the *Data Protection Guide* for detailed information about configuring clusters in a MetroCluster configuration and using SyncMirror to mirror volumes, respectively.

The following figure illustrates the cluster configuration.



Configuration requirements for distances between 500 meters and 10 kilometers

The MetroCluster configurations require the following:

- F825 or F880 cluster pair configured for mirrored volume use
- The following licenses:
  - Cluster—Enables the two nodes to communicate and take over each other's functions in the event of a failure.
  - Syncmirror_local—Enables mirrored volumes.
  - Cluster_remote—Enables mirrored volumes over long distances.
- Two Brocade Silkworm 3200 or 3800 switches that have the following:
  - Firmware version 3.0.2f or later
  - QuickLoop license
  - Long distance Small Form-factor Pluggable (SFP) transceivers

Note

You cannot use a 3800 switch and a 3200 switch in the same MetroCluster configuration. That is, both switches you use must be 3800 switches or 3200 switches.

Configuration limitations for distances between 500 meters and 10 kilometers

The MetroCluster configurations have the following limitations:

- The maximum total capacity is limited to 8 TB.
- The switches used in a MetroCluster configuration cannot be used by Fibre Channel tape devices. Only filers and disk shelves can be connected to the switches in a MetroCluster configuration.
- A tape storage area network (SAN) cannot be connected to either of the filers in a MetroCluster configuration.

Recognizing a disaster

Failures that require disaster recovery

The disaster recovery procedure is an extreme measure that should be used only if the failure disrupts all communication from one MetroCluster site to the other for a prolonged period of time.

The following are examples of disasters that could cause such a failure:

- Fire
- Earthquake
- Prolonged power outages at a site
- Prolonged loss of connectivity from clients to the filers at a site

Ways to determine whether a disaster occurred

It is critical that you follow some predefined methodical procedure to confirm that a disaster really occurred. The procedure should include determining the status of the supposed disaster site by

- Using external interfaces to the filer, such as the following:
  - Ping
  - Remote shell
  - FilerView
- Using network management tools to verify connectivity to the disaster site
- Physically inspecting the disaster site, if possible

You should declare a disaster only after verifying that a disaster occurred and that service cannot be restored.

Failures that do not require disaster recovery

If the failure is one in which the cluster relationship is reestablished after fixing the problem, you do not want to perform the disaster recovery procedure.

Do not perform the disaster recovery procedure for the following failures:

- A failure of the cluster interconnect between the two sites. This can be caused by the following failures:
  - Failure of the interconnect cable
  - Failure of one of the VI cluster adapters
  - If using switches, a failure of the SFP connecting a cluster node to the switch

  With this type of failure, both nodes of the cluster remain running. Automatic takeover is disabled because Data ONTAP cannot synchronize

the nonvolatile RAM (NVRAM) logs. After you fix the problem and reestablish the connection, the cluster nodes resynchronize their NVRAM logs and the cluster returns to normal operation.

- The storage from one site (site A) is not accessible to the cluster node at the other site (site B). This can be caused by the following failures:
  - Failure of any of the cables connecting the storage at one site to the cluster node at the other site or switch
  - If using switches, failure of any of the SFPs connecting the storage to the switch or the cluster node to the switch
  - Failure of the Fibre Channel adapter on the cluster node
  - Failure of a storage disk shelf (loop resiliency circuit, power, access to disk shelves, and so on)

  With this type of failure, you see a “mailbox disk invalid” message on the console of the filer that cannot see the storage. After you fix the problem and reestablish the connection, the cluster returns to normal operation.
- If you are using switches, the Inter-Switch Link (ISL) between the two switches fails.

  With this type of failure, both nodes of the cluster remain running. You see a “mailbox disk invalid” message because a filer at one site cannot see the storage at the other site. You also see a message because the two filers cannot communicate with each other. After you fix the problem and reestablish the connection, the cluster nodes resynchronize their NVRAM logs and the cluster returns to normal operation.

Recovering from a disaster

What to do

After determining that the failure to the MetroCluster configuration is a disaster, you should

- Ensure that the surviving filer is isolated from its partner. See "Restricting access to the disaster site cluster node" on page 505.
- Force the surviving filer to take over the functions of it partner. See "Forcing a node into takeover mode" on page 506.
- Remount the failed partner's volumes. See "Remounting volumes of the failed filer" on page 507.
- Fix the cluster at the disaster site. See "Fixing failures caused by the disaster" on page 507.
- Re-create the MetroCluster configuration. See "Reestablishing the MetroCluster configuration" on page 507.

Restricting access to the disaster site cluster node

You must restrict access to the disaster site cluster node to prevent the cluster node from resuming service. If you do not restrict access, you risk the possibility of data corruption.

You can restrict access to the disaster site cluster node in the following ways:

- Turn off power to the disaster site cluster node
- Use manual fencing

Turning off power to the disaster site cluster node: To turn off power to the disaster site cluster node, complete the following step.

Step	Action
1	Switch off the power at the back of the filer.

This is the best method for restricting access to the disaster site cluster node. You can perform this task at the disaster site or remotely, if you have that capability. Network Appliance recommends this method.

Using manual fencing: You can use manual fencing as an alternative to turning off power to the disaster site cluster node. The manual fencing method restricts access using software and physical means.

To manually fence off the disaster site cluster node, complete the following steps.

Step	Action	
1	Disconnect the cluster interconnect and Fibre Channel adapter cables of the cluster node at the surviving site.	
2	**If you are using...**	**Then fencing is achieved by...**
	Application failover	Using any application specified method that either prevents the application from restarting at the disaster site or prevents the application clients from accessing the application servers at the disaster site. Methods can include powering down the application server, removing an application server from the network, or any other method that prevents the application server from running applications.
	IP failover	Using network management procedures to ensure that the filers at the disaster site are isolated from the external public network.

Forcing a node into takeover mode

To force the surviving cluster node into takeover mode, complete the following step.

Step	Action
1	Enter the following command on the surviving cluster node: `cf forcetakeover -d` **Result:** Data ONTAP causes the following to occur: ◆ The surviving cluster node takes over the functions of the failed partner. ◆ The mirrored relationships between the two plexes of mirrored volumes are broken, thereby creating two unmirrored volumes. This is called splitting the mirrored volumes. The overall result of using the `cf forcetakeover -d` command is that a cluster node at the surviving site is running in takeover mode with all the data in unmirrored volumes.

Remounting volumes of the failed filer

You must remount the volumes of the failed filer because the volumes will be accessed through the surviving filer. To remount the volumes, complete the following steps.

Step	Action
1	On an NFS client at the surviving site, create a directory to act as a mount point. **Example:** `mkdir /n/toaster/home`
2	Mount the volume. **Example:** `toaster:/vol/vol0/home /n/toaster/home`

See the *Software Setup Guide* for details about mounting volumes.

Fixing failures caused by the disaster

Fix the failures caused by the disaster, if possible. For example, if a prolonged power outage to one of the MetroCluster sites caused the failure, the disaster is over after you restore power.

To fix failures, complete the following steps.

Step	Action
1	Ensure that the filer and disk shelves have power.
2	Ensure that all Fibre Channel adapters, VI cluster adapters, cables, and switch ports are connected and working.

You cannot fix failures if the disaster is one in which the site is destroyed. For example, a fire or an earthquake could destroy one of the MetroCluster sites. In this case, you create a new MetroCluster configured partner at a different site.

Once the filer at the surviving site can see the disk shelves at the disaster site, Data ONTAP renames the mirrored volumes that were split by adding a number in parenthesis to the volume name. For example, if the volume name was vol1 before the disaster and the split, the renamed volume name could be vol1(1).

Reestablishing the MetroCluster configuration

Depending on the state of a mirrored volume before you forced the surviving cluster to take over its partner, you use one of two procedures to reestablish the MetroCluster configuration:

- If the mirrored volume was in a normal state before the forced takeover, you can rejoin the two volumes to reestablish the MetroCluster configuration. This is the most typical case.
- If the mirrored volume was in a level-0 resynchronization state before the forced takeover, you cannot rejoin the two volumes. You must re-create the synchronous mirror to reestablish the MetroCluster configuration.

Rejoining the two volumes: To return the cluster to normal operation by rejoining the two volumes, complete the following steps.

Step	Action
1	Validate that you can access the remote storage by entering one of the following commands: **`sysconfig -v`** or **`vol status -r`**
2	Turn on power to the cluster node at the disaster site. **Result:** After the cluster node at the disaster site boots up, it displays the following: `Waiting for Giveback...`
3	Enter the following command at the partner cluster node: **`cf giveback`** **Result:** The cluster node at the disaster site reboots.
4	Determine which volumes are at the surviving site and which volumes are at the disaster site by entering the following command: **`vol status -r`**
5	Take the disaster site's volumes offline by entering the following command for each volume that was split: **`vol offline`** ***`disaster_vol`*** *disaster_vol* is the name of the volume at the disaster site. **Note** An error message appears if the volume is already offline.

Step	Action
6	Re-create the mirrored volumes by entering the following command for each volume that was split: **`vol mirror volume_name -v disaster_vol`** *volume_name* is the volume on the surviving site's cluster node. *disaster_vol* is the volume on the disaster site's cluster node. **Result:** The *volume_name* volume rejoins the *disaster_vol* volume to reestablish the MetroCluster configuration.

For an example of rejoining volumes, see "Example of rejoining volumes" on page 510.

Re-creating the synchronous mirror: To return the cluster to normal operation by re-creating the synchronous mirror, complete the following steps.

Step	Action
1	Validate that you can access the remote storage by entering one of the following commands: **`sysconfig -v`** or **`vol status -r`** **Note** A `(level-0 resync in progress)` message indicates that a plex cannot be rejoined.
2	Turn on the power to the cluster node at the disaster site. **Result:** After the cluster node at the disaster site boots up, it displays the following: `Waiting for Giveback...`
3	Enter the following command at the partner cluster node: **`cf giveback`** **Result:** The cluster node at the disaster site reboots.

Step	Action
4	Destroy every target plex that is in a level-0 resync state by entering the following command: **`vol destroy plex_name`** For details about synchronous mirroring, see the *Data Protection Guide*.
5	Re-create the mirrored volumes by entering the following command for each volume that was split: **`vol mirror volume_name`** For details about synchronous mirroring, see the *Data Protection Guide*.

For an example of rejoining volumes, see "Example of re-creating the synchronous mirror" on page 511.

Example of rejoining volumes

The following example shows the commands and status output when rejoining volumes to reestablish the MetroCluster configuration.

First, the volume status of the disaster site's storage after reestablishing access to the cluster partner at the surviving site is shown.

```
filer1> vol status -r
Volume mir (online, normal) (zoned checksums)
  Plex /mir/plex5 (online, normal, active)
    RAID group /filer1/plex5/rg0 (normal)

RAID Disk Device HA  SHELF BAY CHAN  Used (MB/blks) Phys (MB/blks)
--------- ------ ------------- ----- -------------- --------------
parity   8a.2   8a   0     2   FC:B  34500/70656000 35003/71687368
data     8a.8   8a   1     0   FC:B  34500/70656000 35003/71687368

Volume mir(1) (failed, out-of-date) (zoned checksums)
  Plex /mir(1)/plex1 (offline, normal, out-of-date)
    RAID group /mir(1)/plex1/rg0 (normal)

RAID Disk Device HA  SHELF BAY CHAN  Used (MB/blks) Phys (MB/blks)
--------- ------ ------------- ----- -------------- --------------
parity   6a.0   6a   0     0   FC:B  34500/70656000 35003/71687368
data     6a.1   6a   0     1   FC:B  34500/70656000 35003/71687368

  Plex /mir(1)/plex5 (offline, failed, out-of-date)
```

Next, the mirror is reestablished using the `vol mirror -v` command.

Note

The filer at the surviving site is called filer1; the filer at the disaster site is called filer2.

```
filer1> vol mirror mir -v mir(1)
This will destroy the contents of mir(1).  Are you sure? y
Mon Nov 18 15:36:59 GMT [filer1:
raid.mirror.resync.snapcrtok:info]: mir: created mirror
resynchronization snapshot mirror_resync.1118153658(filer2)
Mon Nov 18 15:36:59 GMT [filer1: raid.rg.resync.start:notice]:
/mir/plex6/rg0: start resynchronization (level 1)
Mon Nov 18 15:36:59 GMT [filer1: raid.mirror.resync.start:notice]:
/mir: start resynchronize to target /mir/plex6
```

After the volumes rejoin, the synchronous mirrors of the MetroCluster configuration are reestablished.

```
filer1> vol status -r mir
Volume mir (online, mirrored) (zoned checksums)
  Plex /mir/plex5 (online, normal, active)
    RAID group /mir/plex5/rg0 (normal)

RAID Disk Device HA  SHELF BAY CHAN  Used (MB/blks) Phys (MB/blks)
--------- ------ ------------- ----- -------------- -------------
parity   8a.2   8a   0     2   FC:B  34500/70656000 35003/71687368
data     8a.8   8a   1     0   FC:B  34500/70656000 35003/71687368

  Plex /mir/plex6 (online, normal, active)
    RAID group /mir/plex6/rg0 (normal)

RAID Disk Device HA  SHELF BAY CHAN  Used (MB/blks) Phys (MB/blks)
--------- ------ ------------- ----- -------------- -------------
parity   6a.0   6a   0     0   FC:B  34500/70656000 35003/71687368
data     6a.1   6a   0     1   FC:B  34500/70656000 35003/71687368
```

Example of re-creating the synchronous mirror

The following example shows the commands and status output when re-creating volumes to reestablish the MetroCluster configuration.

The volume status of the disaster site's storage after reestablishing access to the cluster partner at the surviving site is shown.

```
filer1>vol status -r
Volume mir1 (online, normal) (zoned checksums)
  Plex /mir1/plex0 (online, normal, active)
  RAID group /mir1/plex0/rg0 (normal)
```

```
RAID Disk Device HA  SHELF BAY CHAN  Used (MB/blks) Phys (MB/blks)
--------- ------ ------------- ----- -------------- -------------
parity   8a.3   8a   0     3   FC:B  34500/70656000 35003/71687368
data     8a.4   8a   0     4   FC:B  34500/70656000 35003/71687368
data     8a.6   8a   0     6   FC:B  34500/70656000 35003/71687368
data     8a.5   8a   0     5   FC:B  34500/70656000 35003/71687368

Volume mir1(1) (failed, partial) (zoned checksums)
  Plex /mir1(1)/plex0 (offline, failed, inactive)

  Plex /mir1(1)/plex6 (online, normal, resyncing)
    RAID group /mir1(1)/plex6/rg0 (level-0 resync in progress)

RAID Disk Device HA  SHELF BAY CHAN  Used (MB/blks) Phys (MB/blks)
--------- ------ ------------- ----- -------------- -------------
parity    6a.6   6a  0     6   FC:B  34500/70656000 35003/71687368
data      6a.2   6a  0     2   FC:B  34500/70656000 35003/71687368
data      6a.3   6a  0     3   FC:B  34500/70656000 35003/71687368
data      6a.5   6a  0     5   FC:B  34500/70656000 35003/71687368
```

The mir1(1)/plex6 plex shows that a level-0 resynchronization was in progress; therefore, an attempt to rejoin the plexes fails, as shown in the following.

```
filer1> vol mirror mir1 -v mir1(1)
vol mirror: Illegal mirror state for volume 'mir1(1)'
```

Because the mir1(1)/plex6 plex had a level-0 resynchronization in progress, the mir1(1) volume must be destroyed and the mir volume remirrored to reestablish a synchronous mirror, as shown in the following.

```
filer1> vol mirror mir1 -v mir1(1)
vol mirror: Illegal mirror state for volume 'mir1(1)'
filer1> vol destroy mir1(1)
Are you sure you want to destroy this volume? y
Volume 'mir1(1)' destroyed.
filer1> vol mirror mir1
Creation of a mirror plex with 4 disks has been initiated.  The
disks need to be zeroed before addition to the volume.  The process
has been initiated and you will be notified via the system log as
disks are added.
```

Glossary

ACL Access Control List. A list that contains the users' or groups' access rights to each share.

active file system A file system excluding its snapshots.

blocking factor The number of tape blocks that are transferred in each write operation.

CIFS Common Internet File System. A protocol for networking PCs.

console A terminal that is attached to a filer's serial port and is used to monitor and manage filer operation.

DNS Domain Naming Service. An Internet service for finding IP addresses.

dump path A path that specifies one volume, qtree, or subtree to back up.

file mark Data on a tape that signals a boundary of a tape file.

HTTP HyperText Transfer Protocol. An Internet Web transfer protocol.

increment chain A series of incremental backups of the same path.

inode A data structure containing information about files on a filer and in a UNIX file system.

local tape device A program-based functionality associated with a tape drive that is directly attached to a filer that is performing a tape operation.

MD5 Message Digest 5. A checksum algorithm described in RFC (Request For Comments) 1321, a proposal for an Internet standard.

mirror A volume that contains a read-only copy of data in the active file system of another volume, usually on another filer.

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.

NFS Network File System. A protocol for networking UNIX-based computers.

NIS Network Information Service, formerly called Yellow Pages. An administrative database for networks.

NVFAIL Software that warns you of compromised database validity and automatically renames the database so that it does not restart automatically.

NVRAM 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.

Open Systems platform A non-filer system, such as a server running Solaris, HP-UX, Windows NT, or Windows 2000, whose data can be backed up to a SnapVault secondary storage system.

Open Systems SnapVault agent A software module that can be installed on a non-filer system and enables that system to back up its data to a SnapVault secondary storage system.

plex A physical copy of a file system. An unmirrored volume has one plex; a mirrored volume has two identical plexes.

primary storage system A system whose data is to be backed up by SnapVault.

quota A limit placed on a file system that restricts disk space usage by files with a given UID or GID.

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.

Remote Shell A program that enables a user on one system to execute a program on another system. Remote Shell connections are usually not interactive.

remote tape device A program-based functionality associated with a tape drive that is not directly attached to a filer that is performing a tape operation.

root volume The volume that contains information that controls the entire filer, usually in the /etc/rc file.

secondary storage system A filer or NearStore system to which data is backed up by SnapVault.

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.

SnapMirror Software that performs automated file system replication of a volume onto the same or a separate disk or filer.

SnapRestore Software that restores an entire volume to the state recorded in a previously taken snapshot.

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 enable administrators to back up the filer to tape while the filer is in use.

snapshot reserve The portion of a volume's disk space that is reserved for snapshots.

source filer The filer from which you are replicating data.

subtree A directory in a volume or qtree.

tape block 1,024 bytes of data.

tape device A specific functionality of a physical tape drive that you create by specifying information in the tape device name when you install a tape drive or tape stacker.

tape file Data on a tape delimited by file marks.

tape library Hardware that can access tape cartridges randomly.

tape medium changer Hardware that can load and unload tape cartridges.

tape stacker Hardware that can access tape cartridges from a stack.

vfiler A virtual filer created using the MultiStore™ software that allows you to partition the storage and networking resources of a single filer so that it appears as multiple filers on the network.

volume A file system.

volume copy A way of copying both data in the active file system and data in snapshots from one volume to another.

Index

Symbols

/dev/mc*n* file 429
/etc/dumpdates file 363
/etc/hosts.equiv file 251
/etc/log/backup file 456
/etc/log/snapmirror 126
/etc/nvfail_rename file 464
/etc/snapmirror.allow file
    purpose 89
    sample 90
/etc/snapmirror.conf file
    arguments field defaults 98
    dest_filer field 94
    format 93
    purpose 92
    sample 97
    SnapMirror scheduling in 110
/etc/tape_config directory, contents 321
/etc/tape_config files 321

A

access to remote data 76
ACLs (Access Control Lists)
    including in tape backups 336
    omitting from tape backup 359
    omitting from tape restores 413
aliases, tape
    on multiple filers 308
    purposes for 303
aliasing commands. *See* storage (aliasing) commands
automatic confirmations, defined 412

B

backup terminate command (terminates a dump) 384
backups
    comparison of methods 333
    copying from tape with restore command 397
    creating snapshot_for_backup file for 336
    Data ONTAP process for 336
    defined 332
    estimating size of 339
    event logging for 455
    file information backed up in 335
    full 333
    incremental 333
        after SnapRestore use 70
        disadvantages of 346
        restoring incremental 416
        specifying with dumpdates file 363
    levels defined 348
    metadata 335
    methods Data ONTAP supports 333
    nonconsecutive, contents of 349
    on-line
        advantages 7
        disadvantages 7
        methods 7
    parallel 342
    restarting 383
    restoring data from tapes 388
    restoring files from multiple tapes on single-tape drive, example 423
    restoring files from multiple tapes on tape stacker, example 424
    restoring files from multiple tapes on two single-tape drives, example 423
    restoring files from remote filer, example 422
    restoring files from tape drive on a Solaris system, example 422
    restoring from tape
        displaying a table of contents (files) 401
        displaying a table of contents (qtrees) 401
        ignoring inode limitations 410
        omitting ACLs 413
        omitting qtree information 414
        preparing for other than original path 392
        restoring entire filer to different location, example 419
        restoring entire filer to original location, example 419

restoring entire filer, procedure 417
restoring individual files, example 421
running a test restore 415
specifying a full restore 399
specifying a restore destination 407
specifying a resumed restore, procedure 402
specifying a resumed restore, rules 402
specifying a tape file 406
specifying automatic confirmations 412
specifying individual files to restore 398
specifying restore status 409
specifying tape devices 404
when not to restore from tape 386
when to restore from tape 386
to tape
advantages of 7
disadvantages 9
reasons for 332
unattended 340
types of 336
using df to estimate size of 339
using dump command. *See* backups to tape (dump command)
using SnapMirror for 76
why necessary 2
Windows NT ACL information, including 336
backups to tape (dump command)
appending backups on one tape 365
benefits of entering at console 341
benefits of using Remote Shell 341
blocking factor in 360
estimating tapes required for 340
example 344
excluding data in qtrees 356
excluding files 357
file information backed up 335
format 335
from existing snapshots 337
full backup excluding specified files, example 378
full backup from a snapshot, example 374
full backup of a list of files or directories, example 375
full backup specifying blocking factor, example 377
full backup to local tape minus qtrees, example 368
full backup to local tape with record, example 367
full backup to local tape, example 367
full backup to multiple tapes on single-tape drives, example 372
full backup to multiple tapes on tape stacker, example 372
full backup to remote Solaris system, example 370
full backup to remote tape, example 370
full backup to Solaris system with tape file size limit, example 371
full backup without ACLs, example 376
incremental backup to local tape, example 369
insufficient tapes for 340
labeling backup tapes 347
leaving volumes online for backup 336
maximum tape blocks per tape file 362
minimizing tapes used for 347
number of simultaneous dumps possible 342
objects backed up 335
options for 344
order of tape devices specified by 351
organizing efficiently 346
preparing for 338
qtree batch files in dump paths 354
recommendations for 346
rules for 343
rules for excluding files from 357
simultaneous dumps 337
specifying backup level 348
specifying backup names 358
specifying dump path 353
specifying files and directories 355
specifying local tape device names of 351
specifying remote tape device names of 352
specifying tape blocks per tape file 362
syntax 343
updating dumpdates file 364
use of snapshots 336
verifying 379
where to enter command 341

Windows NT ACL information, omitting 359
blocking factor
specifying when backing up data 360
specifying when restoring data 408
bootup, with nvfail enabled 464

C

cascading SnapMirror destinations 151
CIFS attributes, not backed up by dump command 335
cifs shares command (adds shares with virus scanning turned on/off) 485
cifs shares command (turns on/off virus scanning for read-only access for shares) 484
cifs shares command (turns on/off virus scanning for shares) 482
cifs.snapshot_file_folding.enable option 49
clients
accessing snapshots from 22
disabling access to snapshots for 24
snapshot directory invisible from 24
commands
*See also*SyncMirror commands
See also dump command
See also NDMP commands
See also restore commands
See also SnapMirror commands
See also SnapRestore commands
See also snapshot commands
See also SnapVault commands
See also storage (aliasing) commands
See also tape drive configuration commands
See also volume copy commands
cpio 246
df (finds snapshot disk consumption) 40
hostname 94
ls -a (shows snapshot files) 23
ls -lu (shows when snapshots created) 36
maxfiles (sets maximum number of files per volume) 410
mt -eom (positions tape to end of data) 317
mt -fsf (moves tape forward, skipping files) 318
mt -offline (rewinds tape, unloads tape media) 319
mt -rewind (rewinds tape) 319
mt -status (shows information about tape devices and their drives) 320
mt-bsf (moves tape backward, skips files) 318
options backup.log.enable (turns event logging on or off) 462
qtree (displays status of qtrees) 108
quotas on (enables quotas) 140
rsh (with vol copy abort) 265
rsh (with vol copy status) 260
storage alias (assigns tape alias) 307
storage alias (displays tape aliases) 306
storage show (displays tape drive information) 306
storage unalias (removes tape alias) 308
vol copy throttle 264
vol create -m (creates a pair of mirrored volumes) 276
vol destroy command (removes a plex from a mirrored pair) 290
vol split command (splits a mirrored volume) 288
vol verify start command (compares data in mirrored volumes) 291
vol verify status command (displays percentage of comparison of data in mirrored volumes completed) 293
vol verify stop command (stops comparing data in mirrored volumes) 292
vol verify suspend command (suspends comparison of data in mirrored volumes) 293
compression type
displaying 312
specifying for tape devices 297
specifying in restores from tape 404
configuration files
/etc/nvfail_rename 464
/etc/snapmirror.conf 92
avoiding reversion of 58
confirmations, automatic
defined 412
specifying 412
console, using dump command through 341

copy operations, number of simultaneous allowed 255
copying volumes. *See* volume copy
cpio command
    disadvantages of 246
    using for backup 333

D

data
    access to remote, using SnapMirror 76
    backups, defined 332
    online recovery 6
    protection methods
        defining 2
        need for 2
Data Management Application (DMA) 426
database file protection
    nvfail warnings 464
    renaming files 464
databases, testing using SnapMirror 148
destination volumes, maximum allowed by SnapMirror 77
devices, tape. *See* tape devices
df command
    displaying free space on disk 40
    finding size of data for backups 339
directories of snapshots
    on CIFS clients 23
    on NFS clients 22
disaster recovery using SnapMirror 149
disaster recovery, using SnapMirror for 75
disk pool, defined 270
disk space
    freeing by deleting snapshots 50
    how calculated by snap list 42
    how snapshots consume 37
    percentages used, how calculated 42
    reclaiming from snapshots 47
    reserving for snapshots 45
    saving with file folding 49
DMA. *See* Data Management Application 426
drive number, specifying for tape devices 297
dump command
    appending backups on one tape 365
    backup levels, defined 348
    CIFS attributes, not backed up by 335
    deleting a restartable dump 384
    effect on SnapMirror update schedule 167
    example 344
    excluding data in qtrees from backups 356
    excluding files from backups 357
    full backup excluding specified files, example 378
    full backup from a snapshot, example 374
    full backup of a list of files or directories, example 375
    full backup specifying blocking factor, example 377
    full backup to local tape minus qtrees, example 368
    full backup to local tape with record, example 367
    full backup to local tape, example 367
    full backup to multiple tapes on single-tape drives, example 372
    full backup to multiple tapes on tape stacker, example 372
    full backup to remote Solaris system with tape file size limit, example 371
    full backup to remote Solaris system, example 370
    full backup to remote tape, example 370
    full backup without ACLs, example 376
    how it works 336
    incremental backup to local tape, example 369
    leaving volumes online for 336
    maximum tape blocks per tape file 362
    objects backed up 335
    options for 344
    order of tape devices specified by 351
    permissions limitations 335
    preparing for use 338
    qtree batch files in dump paths 354
    restarting 383
    rules for 343
    rules for excluding files from backups 357
    simultaneous dumps 337
    specifying backup level 348
    specifying backup names 358

specifying dump path 353
specifying files and directories 355
specifying local tape device names 351
specifying remote tape device names of 352
specifying tape blocks per tape file 362
syntax 343
updating dumpdates file 364
use of snapshots 336
using for backups
labeling backup tapes 347
minimizing tapes used for 347
organizing efficiently 346
verifying 379
where to enter 341
Windows NT ACL information, including 336
Windows NT ACL information, omitting 359
with blocking factor 360
dump commands
backup terminate (terminates a dump) 384
dump restart (restarts interrupted backups) 383
dump event log, example 459
dump events 458
dump paths, specifying 353
dump restart command (restarts interrupted backup) 383
dumpdates file
example 363
purpose 363
rules for 363
updating 364

E

error messages
from volume copying 259
Stale File Handle 61
event log files
effect of takeover mode on 456
rotation 456
viewing, reasons for 456
event log messages
aborted restore, example 461
dump events 458
dump, example 459
format 457
identifier field 457
restore events 460
restore, example 460
start and stop logging events 458
timestamp field 457
type field 457
event logging
defined 456
enabling or disabling 462
for backup and restore 455
examples
dump command 344
full backup excluding specified files 378
full backup from a snapshot 374
full backup of a list of files or directories 375
full backup specifying blocking factor 377
full backup to local tape 367
full backup to local tape minus qtrees 368
full backup to local tape with record 367
full backup to multiple tapes on single-tape drives 372
full backup to multiple tapes on tape stacker 372
full backup to remote Solaris system 370
full backup to remote tape 370
full backup to Solaris system with tape file size limit 371
full backup without ACLs 376
incremental backup to local tape 369
dumpdates file 363
event log
aborted restore 461
restore 460
restore command
restoring entire filer to different location 419
restoring entire filer to original location 419
restoring files from multiple tapes on single-tape drive 423
restoring files from multiple tapes on tape stacker 424
restoring files from multiple tapes on two single-tape drives 423

restoring files from remote filer 422
restoring files from tape drive on a Solaris system 422
restoring individual files 421
syntax 394
SnapMirror log files 127
SnapMirror, use of snapshots 114
snapmirror.conf file 97
tape configuration file 322
tape device names 298
vol copy throttle command (controls volume copy speed) 264
volume copy start command 258

F

file access times of snapshots
defined 34
from CIFS clients 36
from NFS clients 36
file extensions
adding to extensions to exclude from a scan 479
adding to vscan list 475
excluding 477
for vscan, excluding 477
for vscan, viewing 475
removing from a vscan exclude list 479
removing from vscan 476, 482
replacing extensions to exclude from a scan 478
replacing in vscan list 476
resetting to vscan default exclude list 479
resetting to vscan default list 476
viewing extensions to exclude from a scant 478
file folding
defined 49
disabling 49
starting 49
file versions of snapshots
defined 34
finding all 34
filers
adding Fiber-Channel-attached drives dynamically 310
creating trust relationships between 251
displaying compression capacity of tape drives 312
displaying information about tape drive connections to 313
displaying SCSI IDs of tape drives 312
displaying tape drive information 312
restoring all volumes on from backup 417
showing which tape drives supported 301
files
backing up using dump command 335
displaying table of contents of for restoration 401
dumpdates 363
event log 456
excluding from dump command 357
nvfail_rename 467
protecting with nvfail option 464
restoring all from tape 399
restoring from snapshots 26
restoring specified files from tape 398
reverting from snapshots with SnapRestore 66
tape qualification 321
using Remote Shell for backups 355
firewalls, how SnapMirror negotiates 85
full backups 333
full restores 399

H

hostname command 94
hosts, remote 296
hourly snapshots 28
HP-UX
backing up with SnapVault 174
versions supported for SnapVault 180
HP-UX. *See also* Open Systems storage platform

I

increment chains, of backups 349
incremental backups
nonconsecutive, contents of 349
initializing SnapMirror 104
inodes
defined 410

displaying for restored files 397
ignoring limits when restoring files 410

L

levels of backups 348
license add
for SnapVault 184
license add command 197
load balancing, using SnapMirror for 76
log file of SnapMirror transfers 126
log files
for backup and restore events 456
NDMP 447
LUN (logical unit number) 304
data protection of volumes containing LUNs 18

M

maxfiles command (sets maximum number of files per volume) 410
metadata backups 335
methods of backup 333
mirrored volumes
adding disks to 282
adding synchronous mirroring to an existing volume 279
advantages 270
creating at the same time 275
defined 269
removing and destroying a plex 290
removing mirroring between 288
verifying data 291
mt command (controls tape movement on tape drives) 316
mt command syntax 316

N

NCS. *See* newest common snapshot
NDMP
advantages of 427
copying with local tool (ndmpcopy) 450
debug log file, displaying 447
debug messages 446
defined 426
disabling preferred network interface 442
disabling service (ndmpd off) 433
enabling or disabling service 433
enabling service (ndmpd on) 433
killing sessions (ndmpd kill command) 449
log files 447
preparing a filer for basic management 431
session information
displaying (ndmpd status command) 443
displaying detailed (ndmpd probe) 444
types 443
setting preferred network interface 441
showing max version supported (ndmpd version command) 439
tape devices used for 429
terminating session (ndmpd kill) 449
Unicode format, file names in 428
using with tape libraries 429
versions of 440
NDMP commands
ndmp on 433
ndmpcopy (uses local copy tool) 450
ndmpd debug (outputs debug log file) 447
ndmpd kill (terminates NDMP session) 449
ndmpd kill (terminates session) 449
ndmpd off (disables service) 433
ndmpd on (enables service) 433
ndmpd probe (displays detailed status) 444
ndmpd status (displays status) 443
ndmpd version (shows max version supported) 439
ndmp.preferred_interface disable option (disables preferred network interface) 442
ndmp.preferred_interface option (sets preferred network option) 441
ndmpcopy command 450
ndmpd debug command (outputs debug log file) 447
ndmpd kill command (terminates NDMP session) 449
ndmpd on/off command 433
ndmpd probe command (displays detailed status) 444
ndmpd status command (displays status) 443

ndmpd version command (shows max version supported) 439
newest common snapshot 84, 144
NFS clients
  accessing snapshots from 22
  determining snapshot file access time from 36
  using cpio command with 333
nightly snapshots 27
nonqualified tape drives
  defined 325
  emulating qualified tape drives 326
nvfail option
  bootup process 464
  renaming files for database protection 467
  sending warnings for database protection 464
  using nvfail_rename file 467
  what it does 464
nvfail_rename file 467
nvpmgr command 196

O

online data recovery 6
Open Systems
  license add command 197
  managing 196
  nvpmgr command 196
  primary platform configuration 190
  restoring data 195
  secondary storage configuration 192
  snapvault snap -x command 198
  snapvault start command 198
  supported commands 196
  svrestore command 197
Open Systems platform
  definition 172
  non-filer systems supported for SnapVault backup 174
  supported platforms 180
Open Systems SnapVault agent
  definition 172
operation numbers
  using with vol copy abort 265
  using with vol.copy.throttle 264
options
  cifs.snapshot_file_filding.enable 49
  license add 184
  ndmpd.preferred_interface (sets preferred network interface) 441
  ndmpd.preferred_interface disable (disables preferred network interface) 442
  snapmirror.checkip.enable (lets SnapMirror resolve host names to IP addresses) 91
  snapmirror.enable off (disables SnapMirror) 137
  snapmirror.enable on (enables SnapMirror) 100
  snapmirror.log.enable 126
  snapvault.access 184, 203
  snapvault.enable off 244
  snapvault.enable on (turns SnapVault on) 184, 185, 192, 202
  vol.copy.throttle (controls volume copy speed) 263
options backup.log.enable command (turns event logging on or off) 462

P

parallel backups 342
permissions limitations, in dump and restore 335
permissions, for snapshots 22
physical path names (PPNs) 303
plexes
  defined 269
  verifying data 291
PPNs (physical path names)
  definition 303
  example 304
  format 304
primary storage system
  defined 172
  filer and Open Systems platforms 174
  supported platforms 180
primary system qtree
  definition 172
private networks, for volume copy 248

Q

qtree command
displaying status of qtrees 108
output when run from vfiler 108
qtrees
backing up data in 356
backing up with dump command 336
batch files
creating 354
specifying dump paths 353
displaying a table of contents for restoring from tape 401
displaying status 108
omitting data in from dump command 356
omitting information from tape restores 414
quotas in SnapMirror 102
qualified tape drives, defined 321
quiesce.*See* snapmirror quiesce command
quotas
enabling on former mirrors 140
quota on command (enables quotas) 140

R

RAID
description of 267
group size
changing (vol volume) 272
RAID commands
vol volume (changes RAID group size) 272
recovery from disaster, using SnapMirror 75
remote access to data, using SnapMirror for 76
remote hosts 296
Remote Shell
using automatic confirmation when restoring from tape 412
using dump command through, benefits of 341
using to display for table of contents for restores from tape 400
using with specified files to dump 355
using with vol copy 254
using with vol copy status 260
renaming snapshots 53
replicating volumes 71, 75
restartable backups
deleting 384
finding when needed 380, 382
qualifications 382
restarting transfers, in SnapMirror 98
restore command
data that cannot be restored 387
disk space required for 390
displaying a table of contents (files) 401
displaying a table of contents (qtrees) 401
for other than original path 392
how it works 388
how to run 397
ignoring inode limitations 410
information required for using 390
omitting ACLs 413
omitting qtree information 414
options 395
permissions limitations 335
preparing for 389
restoring entire filer to different location, example 419
restoring entire filer to original location, example 419
restoring entire filer, procedure 417
restoring files from multiple tapes on single-tape drive, example 423
restoring files from multiple tapes on tape stacker, example 424
restoring files from multiple tapes on two single-tape drives, example 423
restoring files from remote filer, example 422
restoring files from tape drive on a Solaris system, example 422
restoring incremental backups 416
restoring individual files, example 421
rules 394
running a test restore 415
specifying a full restore 399
specifying a restore destination 407
specifying a resumed restore 402
specifying a tape file 406
specifying automatic confirmations 412
specifying individual files to restore 398
specifying restore status 409
specifying tape devices 404

syntax 394
tapes required for 390
types of restores 394, 395
using Remote Shell for displaying table of contents 400
using with Remote Shell 391
when not to use 386
when to use 386
Windows NT ACLs 395
restore event log
aborted restore, example 461
successful restore, example 460
restore events 460
restores
displaying a table of contents (files) 400
displaying a table of contents (qtrees) 401
event logging for 455
ignoring inode limitations 410
incremental 416
omitting ACLs 413
omitting qtree information 414
preparing for 389
restoring entire filer to different location, example 419
restoring entire filer to original location, example 419
restoring entire filer, procedure 417
restoring files from multiple tapes on tape stacker, example 424
restoring files from multiple tapes on two single-tape drives, example 423
restoring files from remote filer, example 422
restoring files from tape drive on a Solaris system, example 422
restoring individual files, example 421
running a test restore 415
specifying a full restore 399
specifying a restore destination 407
specifying a resumed restore, procedure 402
specifying a resumed restore, rules 402
specifying a tape file 406
specifying automatic confirmations 412
specifying individual files to restore 398
specifying restore status 409
specifying tape devices 404
types of 395
resynchronizing SnapMirror source and mirror 145
retrying transfers, in SnapMirror 98
reverting volumes with SnapRestore 61
rewind type, specifying for tape devices 297
rmt protocol 334
root volumes, reverting with SnapRestore 59
rsh command
using with vol copy abort 265
using with vol copy status 260
rules
dump command 343
for dumpdates file 363
for excluding files from dump command 357
for restore command 394
for specifying a resumed restore 402

S

SAN
data protection in the SAN environment 18
schedules
for default snapshots 27
for user-defined snapshots 31
strategies for snapshots 29
where to specify for SnapMirror 92
SCSI ID, of tape drives, displaying 312
secondary storage
supported platforms 180
secondary storage system
definition 172
role in a basic SnapVault deployment 175
secondary system qtree
definition 172
shares
adding shares with virus scanning turned off 485
enabling or disabling vscan 482
enabling or disabling vscan for read-only shares 484
simultaneous transfers 77
single-tape tape drives 372
size, volume, verifying for SnapMirror 250
snap list -o command (lists origins of qtree snapshots) 223

snap list -q command (lists qtree snapshots) 220
snap restore -t file command (reverts file from snapshot) 66
snap restore -t vol command (reverts volume from snapshot) 63
SnapMirror
  /etc/snapmirror.conf file
    arguments field defaults 98
    dest_filer field (defines destination filer) 94
  backing up to tape with 156
  blocking transfers 132
  cascading destinations
    defined 151
    restructuring a cascade 153
  cascading mirrors 151
  checking status (snapmirror status) 116, 125
  complete transfer, when to use 102
  converting mirrors to regular volumes (snapmirror break) 139
  copying mirrors from mirrors 151
  enabling quotas on former mirrors (quota on) 140
  halting a transfer 131
  host names, resolving to IP addresses 91
  license requirement for 71
  log file 126
  logging transfer history 126
  maximum number of destination volumes allowed by 77
  mirroring volumes 75, 100
  names of snapshots created 113
  overview of 75
  preserving quotas after converting from a mirror 139
  releasing mirrors 142
  replicating volumes 100
  replicating volumes to tape (snapmirror store) 156
  requirements 83
  restarting replication of volumes to new tape (snapmirror store) 159, 164
  restarting transfers, defined 98
  restoring volumes from tape (snapmirror retrieve) 160, 164, 165
  restricting destination volumes 103
  restrictions 83
  resuming transfers after quiescing 134
  retrying transfers, defined 98
  schedules, where specified 110
  snapmirror.log.enable option 126
  specifying destinations
    using options snapmirror.access 89
    using snapmirror.allow file 90
  status messages 116, 125
  stopping replication of a volume (snapmirror abort) 131
  tips for efficient scheduling 100
  transfers through firewalls 85
  turning off scheduled updates for filer (snapmirror off) 137
  using for database testing 148
  using for disaster recovery 149
  using with dump command 167
  using with snapshots, example 114
  verifying volume size 250
snapmirror abort command (halts specified transfer) 131
snapmirror break command (converts mirror to regular volume) 140, 150, 166
SnapMirror commands
  options snapmirror.enable off (disables SnapMirror) 137
  options snapmirror.enable on (enables SnapMirror) 100
  snapmirror abort (halts specified transfer) 131
  snapmirror abort (stops replication of a volume) 131
  snapmirror break (converts mirror to regular volume) 140, 150, 166
  snapmirror destinations (lists required snapshots) 152
  snapmirror initialize (performs base-level transfer to mirror) 104
  snapmirror off (disables SnapMirror) 137
  snapmirror off (turns off scheduled updates for filer) 137
  snapmirror on (enables SnapMirror) 100
  snapmirror quiesce (blocks transfers) 132
  snapmirror release (releases a volume from

mirror status) 142
snapmirror resume (resumes transfers after quiescing) 134
snapmirror resync (resynchronizes a source to a mirror) 145
snapmirror retrieve (restores volumes from tape) 160, 164, 165
snapmirror status (checks data replication status) 125
snapmirror status (checks data transfer status) 116
snapmirror store (replicates volume to tape) 156
snapmirror update (performs unscheduled incremental update of mirror) 110
snapmirror use (restarts replication to new tape) 159, 164
snapmirror destinations command (lists required snapshots) 152
snapmirror initialize command (performs base-level transfer to mirror) 104
snapmirror quiesce command (blocks transfers) 132
snapmirror release command (releases a volume from mirror status) 142
snapmirror resume command (resumes transfers after quiescing) 134
snapmirror resync command (resynchronizes a source to a mirror) 145
snapmirror retrieve command (restores volumes from tape) 160, 165
snapmirror store command (replicates volume to tape) 156
snapmirror update command (performs unscheduled incremental update of mirror) 110
snapmirror use command (restarts replication of volumes to new tape) 159, 164
snapmirror.access option (specifies SnapMirror destinations from SnapMirror source) 89
snapmirror.allow file (specifies SnapMirror destinations from SnapMirror source) 90
snapmirror.checkip.enable option (lets SnapMirror resolve host names to IP addresses) 91
snapmirror.conf file. *See* /etc/snapmirror.conf file
snapmirror.log.enable option 126
SnapRestore
avoiding reversion of configuration files 58
effect on snapshots used 58
how it works 56
license requirement for 55
prerequisites for 57
preserving configuration files using 58
reverting root volumes using 59
reverting volumes using (vol snaprestore) 61
typical applications using 56
using with NFS 61
what it does not revert 56
when to use 57
with previous releases 57
SnapRestore commands
snap restore -t file (reverts file from snapshot) 66
snap restore -t vol (reverts volume from snapshot) 63
snapshot basename
definition 172
snapshot commands
snap create (creates snapshots manually) 33
snap delete (deletes snapshots) 51
snap list (shows disk space used by snapshots) 41
snap list -o (lists origins of qtree snapshots) 223
snap list -q (lists qtree snapshots) 220
snap rename (renames snapshots) 53
snap reserve (sets percent of disk space for) 48
snap sched (changes snapshot schedule) 31
snap sched (displays snapshot schedule) 28
vol options nosnapdir (makes directories visible or invisible) 25
snapshots
accessing
from CIFS clients 24
from NFS clients 23
automatic types of 27
changing schedules (snap sched) 31
creating manually (snap create) 33
default schedule 27
defined 20
deleting (snap delete) 51

deleting manually 51
deleting when locked 51
determining which to delete 43
df command (displays disk usage) 40
directories
making visible or invisible 25
on CIFS clients 23
on NFS clients 23
directory structure on NFS clients 22
disk consumption by 34, 50
displaying schedules (snap sched) 28
file access times
defined 34
from CIFS clients 36
from NFS clients 36
file names on NFS clients 23
file versions, defined 34
finding all versions of a file 34
hourly 28
how snap list results are calculated 41
invisible directories 23
ls command (shows files in a directory) 23
making directories visible or invisible (vol options nosnapdir) 25
maximum number of 20
names created by SnapMirror 113
newest common 84
nightly 27
permissions to read 22
reclaiming disk space from 47
renaming (snap rename) 53
reserving space for 41, 45
restoring files from 26
saving disk space with file folding 49
schedules, sample file 31
setting space reserved for 48
showing disk space used (snap list command) 41
showing SnapMirror-required (snapmirror destinations) 152
snap list (shows disk space used by) 41
snapshot_for_backup file 336
strategies for scheduling 29
time created 36
used by SnapMirror, example 114
uses of 21
using with SnapRestore 56
weekly 27
SnapVault
advantages 173
backup for qtrees, ending 237
basic configuration for filers 183
primary storage 184
restoring 188
scheduling 186
secondary storage 184
basic deployment 174
changing configuration 199, 225
checking status (snapmirror status) 214
estimating time for initial baseline backup 182
license add command 184, 197
maximum number of snapshots per volume 181
nvpmgr command 196
objects backed up on filer systems 173
Open Systems configuration 190
configuring the Open Systems platform 190
configuring the secondary storage system 192
managing 196
restoring data 195
supported commands 196
Open Systems platform objects backed up 173
overview of filer backup process 177
overview of Open Systems backup process 179
planning 180
planning backup schedule and snapshot retention 181
planning primary qtrees or directories and secondary qtrees 180
planning to store more than 251 snapshots per volume 182
primary configuring 198
primary to secondary to tape deployment 175
primary, configuring 200, 210
qtree snapshots, listing 220
qtree snapshots, listing origins of 223
restoring data from secondary to primary 231
secondary to SnapMirror deployment 176

secondary, configuring 211
snapshot schedule, unconfiguring 238
snapshots
displaying information about 214
releasing 233, 240
snapvault snap -x command 198
snapvault start command 198
snapvault.access option (controls access) 184, 203
snapvault.enable off option (turns SnapVault off) 244
snapvault.enable on option (turns SnapVault on) 184, 185, 192, 202
status messages 214
supported Open Systems platforms 180
supported platforms 180
svrestore command 197
terminology 172
updating schedules snapshots manually 228
updating secondary qtree 199
updating sercondary qtree 226
snapvault access option 184, 203
SnapVault baseline transfer
definition 173
SnapVault commands
snap sched command (configures primary snapshot schedule) 198, 200, 210
snap sched -x command (configures secondary snapshot schedule) 211
snapvault modify (changes SnapVault configuration) 199, 225
snapvault release (releases unneeded snapshots) 233, 240
SnapVault
snapshots
releasing 240
snapvault restore (restores data from secondary to primary 231
snapvault snap create (manually updates existing snapshot) 228
snapvault snap unsched command (unconfigures SnapVault) 238
snapvault start (initializes SnapVault backup relationship) 206, 232
snapvault status (checks data replication status) 214
snapvault status (displays information about snapvault snapshots) 214
snapvault stop (ends backup for qtrees) 237
snapvault update (updates SnapVault secondary qtree) 199, 226
SnapVault incremental transfer
definition 173
snapvault modify command (changes SnapVault configuration) 199, 225
SnapVault relationship
definition 172
snapvault release command (releases unneeded snapshots) 233, 240
snapvault restore command (restores from secondary to primary 231
snapvault snap create command (manually updates existing snapshot) 228
snapvault snap sched command (configures primary snapshot schedule) 198, 200, 210
snapvault snap sched -x command (configures secondary snapshot schedule) 211
snapvault snap unsched command (unconfigures SnapVault) 238
snapvault snap -x command
supporting Open Systems backup 198
SnapVault snapshot
definition 172
SnapVault snapshot basename
defintion 172
snapvault start command
supporting Open Systems backup 198
snapvault start command (initializes SnapVault backup relationship) 206, 232
snapvault status command (displays information about snapvault snapshots) 214
snapvault stop command (end backup for qtrees) 237
snapvault update command (updates SnapVault secondary qtree) 199, 226
snapvault.enable off option 244
snapvault.enable on option (turns SnapVault on) 184, 185, 192, 202
Solaris
backing up with SnapVault 174

versions supported for SnapVault 180
Solaris ufsrestore command 335
Solaris. *See also* Open Systems storage platform
space on disks, how snapshots consume 37
stackers, tape, backups to 372
storage (aliasing) commands
storage alias (assigns tape alias) 307
storage alias (displays tape aliases) 306
storage show (displays tape drive information) 306
storage unalias (removes tape alias) 308
storage alias command (assigns tape alias) 307
storage alias command (displays tape aliases) 306
storage show command (displays tape drive information) 306
storage show tape supported command (shows acceptable tape drives) 301
storage stats tape command (displays tape statistics) 299
storage stats tape zero command (restarts tape statistics display) 300
storage unalias command (removes tape alias) 308
subtrees, defined 353
supported per filer 77
svrestore command 197
SyncMirror
adding disks to mirrored volumes 282
adding synchronous mirroring to an existing volume 279
creating mirrored volumes 275
removing and destroying a plex 290
removing mirroring between mirrored volumes 288
SyncMirror commands
vol add (adds disk to mirrored volume) 283
vol create -m (creates a pair of mirrored volumes) 276
vol destroy (removes a plex from a mirrored pair) 290
vol mirror (adds mirroring to existing volume) 279
vol split command (splits a mirrored volume) 288
vol verify start command (compares data in mirrored volumes) 291
vol verify status command (displays percentage of comparison of data in mirrored volumes completed) 293
vol verify stop command (stops comparing data in mirrored volumes) 292
vol verify suspend command (suspends comparison of data in mirrored volumes) 293
SyncMirror license
disabling 271
enabling 271
sysconfig -m command (shows information on tape media changers) 313
sysconfig -t command (shows tape drive information) 312
sysconfig -v command (shows tape drive connections to filer) 313

T

tape aliases
definition 303
how to display 305
on multiple filers 308
physical path names (PPNs) 303
tape backup
of a SnapVault secondary 175
tape backups. *See* backups to tape (dump command), SnapMirror, backing up to tape with
tape blocks
specifying the size for dump 360
tape configuration file, example 322
tape devices
defined 295
example of name listing 298
examples of names 298
format of names 296
local, defined 296
on remote Solaris systems 296
remote, defined 296
specifying compression type of 297
specifying for a resumed restore of 404
specifying rewind number of 297
specifying rewind type of 297
specifying when restoring data 396

tape libraries, qualified tape drives in 311
tape stackers, backing up to multiple tapes on 372
use for 296
tape drive configuration commands
sysconfig -m (shows information about tape media changers) 313
sysconfig -t (shows tape drive information) 312
sysconfig -t (shows what tape drives are on a filer) 311
sysconfig -v (shows tape drive connections to filer) 313
tape drive slot on the filer, displaying 312
tape drives
adding Fibre-Channel-attached drives dynamically 310
connections to filer, displaying information about 313
controlling tape movement 316
displaying compression capacity 312
displaying SCSI IDs 312
displaying which ones are on a filer 311
displaying which supported 301
how filer qualifies 323
in tape stackers, listing qualified 311
local, specifying backups on 367
moving tape backward (mt-bsf) 318
moving tape forward, skipping files (mt -fsf) 318
multiple tapes, using for backup on a tape stacker 372
nonqualified
defined 325
displaying information 325
emulating qualified tape drives 326
getting information for emulation 326
information needed for emulation 325
using 311
positioning tape to end of data (mt -eom) 317
qualified
defined 321
list of 311
qualifying nonqualified drives 321
remote, specifying 296
rewinding tape (mt -rewind) 319
showing status (mt -status) 320
single-tape, using multiple tapes on 372
skipping files (mt-bsf) 318
tape media changers, displaying information about 313
unloading tape after rewind (mt -offline) 319
usable by NDMP 429
tape files
compression types 315
definition of 315
maximum size for dump 362
specifying local tape device names of 351
specifying remote tape device names of 352
specifying size for dump of 362
tape libraries
defined 427
qualified tape drives in 311
recognized by filers 429
showing names assigned to 429
using with NDMP 429
tape media changers, displaying information about 313
tape qualification files
creating 321
error messages related to, example 323
format 321
how used by filer 323
list of SCSI command timeout values 321
requirements 321
tape restores
displaying a table of contents (files) 400
displaying a table of contents (qtrees) 401
ignoring inode limitations 410
incremental 416
omitting ACLs 413
omitting qtree information 414
preparing for 389
restoring entire filer to different location, example 419
restoring entire filer to original location, example 419
restoring entire filer, procedure 417
restoring files from multiple tapes on single-tape drive, example 423

restoring files from multiple tapes on tape stacker, example 424
restoring files from multiple tapes on two single-tape drives, example 423
restoring files from remote filer, example 422
restoring files from tape drive on a Solaris system, example 422
restoring individual files, example 421
running a test restore 415
specifying a full restore 399
specifying a restore destination 407
specifying a resumed restore, rules 402
specifying a tape file 406
specifying automatic confirmations 412
specifying individual files to restore 398
specifying restore status 409
specifying tape devices 404
types of 395
tape stackers
backing up to multiple tapes on 372
displaying information about 313
tape statistics
displaying 299
restarting display 300
tape_alias file 308
tapes
backing data up to 331
displaying statistics 299
estimating number needed for backups 340
insufficient number of for backups 340
reasons to use for backups 332
required for restore 390
restarting SnapMirror on multiple 159, 164
restarting tape statistics display 300
restoring data from 388
transfers, simultaneous, supported per filer 77
trusted host 251

U

updating SnapMirror mirror 110

V

vfilers
controlling use of host filer virus scanners 492
definition 492
qtree command output displayed 108
virus scanning
adding extensions of files to exclude from a scan 479
adding extensions of files to scan 475
adding shares with virus scanning turned off 485
checks status of vscan options 493
controlling vfiler use of host virus scanners 492
designating secondary virus-scanning clients 473
displaying extensions of files to exclude from scan 478
enables or disables scanning on read-only shares 484
enabling and disabling 473
enabling and disabling mandatory scanning 491
excluding extensions of files to scan 477
extension exclude list compared to extension include list 480
identifying clients that are scanning 487
obtaining NetApp-supported virus-scanning software 470
removing extensions of files to exclude from a scan 479
removing extensions of files to scan 476, 482
replacing extensions of files to exclude from a scan 478
replacing extensions of files to scan 476
resetting extensions of files to exclude from a scan to default 479
resetting extensions of files to scan to default list 476
resetting scanned files cache 495
resetting the virus scan request timeout 490
setting the virus scan request timeout 490
stopping a scanner session 494
turning scanning off for shares 482
viewing extensions of files to scan 475, 477
vol add command (adds disk to mirrored volume) 283
vol copy throttle command 264

vol create -m command (creates a pair of mirrored volumes) 276
vol destroy command (removes a plex from a mirrored pair) 290
vol mirror command (adds mirroring to existing volume) 279
vol options commands
vol options nvfail off (disables protection) 466
vol options nvfail on (enables protection) 466
vol restrict command, in SnapMirror 103
vol status -b 250
vol verify start command (compares data in mirrored volumes) 291
vol verify status command (displays percentage of comparison of data in mirrored volumes completed) 293
vol verify stop command (stops comparing data in mirrored volumes) 292
vol verify suspend command (suspends comparison of data in mirrored volumes) 293
vol.copy.throttle option (controls volume copy speed) 263
volcopy dump operation (backs up to tape) 255
volcopy restore operation (restores backups from tape) 255
volume commands
vol volume (changes RAID group size) 272
volume copy
advantages 246
arguments for copying snapshots 256
bringing volumes online (vol online) 252
changing copying speed (options vol.copy.throttle) 263
checking status (vol copy status) 260
copying one volume to another (vol copy start) 258
copying volumes (vol copy start) 255
defined 246
dump operation (reads data from source) 255
enabling Remote Shell access for 254
errors during 259
finding size of volume (vol status -b) 250
finding volume status (online or offline) 252
number of simultaneous operations 255
operation numbers 260
private networks for 248
requirements for 248
restore operation (writes data to destination) 255
sample status message from filer 261
sample status message using rsh 261
stopping (vol copy abort) 265
taking volumes offline (vol offline) 252
to same or different filers 248
uses for 246
volume copy commands
examples of arguments to 258
options rsh.enable on (enables Remote Shell) 254
options vol.copy.throttle (changes copying speed) 263
vol copy abort (halts the copy operation) 265
vol copy arguments for copying snapshots 256
vol copy start (copies one volume to another) 258
vol copy status (checks status) 260
vol offline (takes volume offline) 252
vol online (brings volume online) 252
vol status (shows whether online or offline) 252
vol status -b (finding size of volume) 250
volume copy operation numbers
using with vol copy abort 265
using with vol.copy.throttle 264
volume size, verifying for snapmirror 250
volumes
backing up with dump command 336
changing the speed of copying 263
converting from mirror to regular 139
destination, maximum number of 77
differences between mirror and regular 116
disabling SnapMirror replication 137
enabling SnapMirror replication 100
finding number of blocks in 250
replicating 71
replicating with SnapMirror 75, 100
reverting from snapshots with SnapRestore 63
reverting with SnapRestore 61, 65
vscan commands
cifs shares -add (adds CIFS shares with virus

scanning turned off 485
cifs shares -change (enables or disables scanning for a share)
enabling or disabling scanning for read-only shares 484
cifs shares -change (enables or disables virus scanning for a share) 482
vscan extensions (shows extensions of files to scan) 475, 477
vscan extensions add (adds extensions of files to scan) 475
vscan extensions exclude (excludes extensions of files to scan) 477, 478
vscan extensions exclude add (adds extensions of files to exclude from a scan) 479
vscan extensions exclude remove (removes extensions of files to exclude from a scan) 479
vscan extensions exclude reset (resets extensions of files to exclude from a scan to default list) 479
vscan extensions exclude set (replaces extensions of files to exclude from a scan) 478
vscan extensions remove (removes extensions of files to scan) 476, 482
vscan extensions reset (resets extensions of files to scan to default list) 476
vscan extensions set (replaces extensions of files to scan) 476
vscan off (disables virus scanning) 473
vscan on (enables virus scanning) 473
vscan reset (resets scanned files cache) 495
vscan scanners (identifies clients that are scanning) 487
vscan scanners secondary scanners (designates backup virus scanners) 473
vscan scanners stop (stops scanner session) 494
vscan options
use_host_scanners 492
vscan options mandatory_scan (sets mandatory virus scanning) 491
vscan options timeout reset (resets the virus scan request timeout) 490
vscan options timeout set (sets the virus scan request timeout) 490

W

weekly snapshots 27
Windows 2000
backing up with SnapVault 174
platforms supported for SnapVault 180
See also Open Systems storage platform
Windows NT
platforms supported for SnapVault 180
See also Open Systems storage platform
Windows NT ACLs
omitting from restores from tape 413
Wndows NT
backing up with SnapVault 174
world wide names (WWNs) 305
WWNs (world wide names)
definition 305
example 305
format 305