Data ONTAP® 7.1
Data Protection
Online Backup and Recovery 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 210-01081_B0
December 2005

Copyright and trademark information

Copyright information

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Portions Copyright © 2001, Sitraka Inc.

Portions Copyright © 2001, iAnywhere Solutions

Portions Copyright © 2001, i-net software GmbH

Portions Copyright © 1995 University of Southern California. All rights reserved.

Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by the University of Southern California, Information Sciences Institute. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission.

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

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

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

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

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

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

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

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

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS

FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

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

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

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

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

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

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

Trademark information

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

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

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

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

Network Appliance NetCache is certified RealSystem compatible.

Table of Contents

Preface xiii

Chapter 1 **Introduction to Data Protection** 1

Data protection options 2

Online backup and recovery 7

Database protection with the NVFAIL feature 10

Protecting against a data loss disaster 11

Data protection in a SAN environment 15

Chapter 2 **Snapshot Management** 17

Understanding Snapshot copies 18

Accessing Snapshot copies from clients 20

Restoring files from Snapshot copies 24

Creating Snapshot schedules 25

Creating Snapshot copies 32

Finding the Snapshot copy you need 33

Understanding Snapshot disk consumption 36

- Monitoring Snapshot disk consumption 39
- Displaying Snapshot disk consumption statistics 40
- Understanding the Snapshot reserve 44
- Changing the Snapshot reserve 47

Saving disk space by using file folding 48

Displaying data change rate between Snapshot copies 49

Snapshot copy autodelete 52

Displaying space reclaimed from deleted Snapshot copies 55

Deleting Snapshot copies 56

Renaming Snapshot copies 59

Chapter 3 **Data Recovery Using SnapRestore** 61

Understanding SnapRestore 62

Deciding when to use SnapRestore 63
Installing the SnapRestore license 66
Reverting a volume to a selected Snapshot copy 67
Reverting a file to a selected Snapshot copy 70
Obtaining correct incremental backups after reversion. 75

Chapter 4 **Data Protection Using SnapMirror** 77
SnapMirror overview 79
Effects of FlexVol volumes on SnapMirror replication 90
Synchronous SnapMirror 93
SnapMirror commands and configuration files. 96
Considerations when planning and running SnapMirror 100
Setting up a basic SnapMirror operation 105
Specifying destination systems on the source 113
Defining source and destination through snapmirror.conf 116
Using SnapMirror over multiple paths 126
Enabling SnapMirror 128
Initializing a SnapMirror destination 130
Updating a destination manually 138
Listing SnapMirror Snapshot copies 141
Checking SnapMirror data transfer status 144
Checking SnapMirror data transfer logs 153
Aborting a SnapMirror transfer. 158
Stabilizing (quiescing) destinations before a Snapshot copy. 160
Turning off or changing scheduled updates for volumes or qtrees. 163
Turning off SnapMirror updates 166
Converting a destination to a writable volume or qtree 168
Releasing partners from a SnapMirror relationship 171
Resynchronizing SnapMirror. 174
Migrating data between volumes by using SnapMirror 182

Moving volume SnapMirror or qtree SnapMirror sources 184
Moving volume SnapMirror sources 185
Moving qtree SnapMirror sources 187
Copying from one destination to another in a series (cascading) 190
Using SnapMirror to copy a volume to local tape 196
How SnapMirror works with the dump command 208
Fixing changes to SnapMirror elements 209
Creating SnapLock destination volumes 211
Protecting SnapVault secondaries using volume SnapMirror 216

Chapter 5 **Data Protection Using SnapVault** 219
SnapVault overview 220
Planning SnapVault. 228
Setting up a basic SnapVault backup. 231
Setting up SnapVault backup on Open Systems platforms. 239
Configuring the SnapVault secondary storage system 241
Managing SnapVault backup of Open Systems platforms 244
Restoring a directory or file 248
Restoring an entire primary storage system 251
Restoring files to a primary storage system from tape 252
Backing up a database 253
Enabling SnapVault. 254
Starting a SnapVault backup relationship 258
Backing up qtree data 259
Backing up non-qtree data 261
Backing up volume data 263
Scheduling SnapVault Snapshot copy updates 266
Checking SnapVault transfers 272
Displaying SnapVault Snapshot copies. 278
Changing settings for SnapVault backup relationships. 282
Manually updating individual secondary storage system qtrees 284
Creating a Snapshot copy manually 286
Restoring SnapVault data to the primary storage system 288

Aborting SnapVault transfers 293
Ending SnapVault backups for a qtree 295
Unscheduling SnapVault Snapshot copies 296
Releasing SnapVault relationships 297
Using SnapVault to back up data to SnapLock volumes 298
Planning capacity 300
Setting up a SnapVault backup 303
Managing WORM Snapshot copies from SnapVault 309
Managing SnapVault log files 326
Protecting your Compliant SnapVault backups with SnapMirror 333
Protecting your SnapVault backups through SnapMirror 340
Turning SnapVault off 343
VERITAS NetBackup and SnapVault 344
Setting up for NetBackup management 346
Using NetBackup to manage SnapVault qtrees 350
Scheduling efficient backup policies 352
VERITAS NetBackup and NearStore 354
Setting up SnapVault support for NetBackup transfers 357
Managing data transfers 360

Chapter 6 **SnapLock Storage Management** 363
About SnapLock 364
Creating SnapLock volumes 366
Managing the compliance clock 368
Setting volume retention periods 370
Destroying SnapLock volumes and aggregates 373
Managing WORM data 375

Chapter 7 **Volume Copy** 377
Learning about volume copy 378
Preparing to copy a volume 380
Verifying the size of each volume 382
Verifying the relationship between storage systems 384
Verifying and changing status of source and destination volumes . . .385
Enabling remote access 387

Copying volumes 388
Checking the status of a volume copy operation 394
Displaying the current speed for copying a volume 396
Controlling the speed of a volume copy operation 397
Aborting a volume copy operation 399

Chapter 8 **SyncMirror Management** 401
Understanding mirrored aggregates 402
Enabling and disabling the mirroring license. 406
Creating mirrored aggregates and traditional volumes 409
Creating a mirrored aggregate or traditional volume 414
Adding a plex. 420
Adding disks to mirrored aggregate or traditional volumes 423
Changing the state of a plex 428
Splitting mirrored aggregates or traditional volumes. 430
Rejoining split aggregates or traditional volumes 432
Removing and destroying a plex 434
Comparing the plexes of mirrored aggregate or traditional volumes. 437

Chapter 9 **Database Protection Using NVFAIL** 441
Understanding the nvfail option 442
Enabling and disabling database file protection 444
Using the nvfail_rename file for additional database protection 445

Chapter 10 **Virus Protection for CIFS** 447
Understanding CIFS virus protection. 448
Setting up and starting virus scanning 450
Specifying file types to be scanned. 453
Excluding file types to be scanned 455
Specifying shares for scanning 459
Turning scanning on or off for shares 460
Turning scanning on or off for read-only access for shares 462

Adding shares with virus scanning turned on or off 463
Displaying the scanner list 465
Checking vscan information 466
Setting and resetting the virus scan request timeout 467
Allowing file access when the scan cannot be performed 468
Controlling vFiler usage of host storage system virus scanners 469
Checking the status of virus-scanning options 471
Stopping a virus scanner session 472
Resetting the scanned files cache 473
Enabling virus scan messages to CIFS clients 474

Glossary 475

Index 481

Preface

About this guide

This guide describes how to protect, back up, restore, and copy data between storage systems that run Data ONTAP™ 7.1 software and that are Serving Data by Design™.

Audience

This guide is for system administrators who are familiar with operating systems that run on storage system clients, such as UNIX®, Windows 95™, Windows NT®, and Windows® 2000. It also assumes that you are familiar with how to configure the storage system 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.

Terminology

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

FilerView as an alternative to commands

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

Another method of performing common tasks is to use the FilerView® graphical management interface for viewing and managing a storage system from a Web browser. FilerView is easy to use, and it includes Help that explains FilerView features and how to use them.

For more information about accessing a storage system using FilerView, and about FilerView Help, see the *Storage Management Guide*.

Command conventions

You can enter storage system commands either on the system console or from any client computer that can access the storage system 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 th 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 7
- "Database protection with the NVFAIL feature" on page 10
- "Protecting against a data loss disaster" on page 11
- "Data protection in a SAN environment" on page 15

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 data. This guide describes how to use Data ONTAP online features like SnapMirror® technology to protect data. See the *Tape Backup and Recovery Guide* for information about using tape to protect data.

Data protection feature	Description
`aggr copy`	This is fast block copy of data stored in aggregates; it enables you to quickly copy blocks of stored system data from one aggregate to another. For information about aggregates and `aggr copy` see the *Storage Management Guide*.

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 Snapshot copies) of data on a volume. Snapshot copies use only a minimal amount of additional volume space, and do not have a performance cost. Snapshot copies are also used to create clones of FlexVol™ volumes and Data ONTAP LUNs. 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 Snapshot copies taken. See Chapter 2, "Snapshot Management" on page 17. You can also create clones of FlexVol volumes and Data ONTAP LUNs using Snapshot copies. See the *Storage Management Guide* for more details.
SnapRestore® (license required)	Fast, space efficient restoration of large volumes of data backed up to Snapshot copies. The SnapRestore feature performs on-request Snapshot recovery from Snapshot copies on an entire volume. See Chapter 3, "Data Recovery Using SnapRestore" on page 61.

Data protection feature	Description
SnapMirror® (license required)	Volume-to-volume and qtree-to-qtree replication. This feature enables you to periodically make Snapshot copies of data on one volume or qtree, replicate that data to a partner volume or qtree, usually on another storage system, and archive one or more iterations of that data as Snapshot copies. Replication on the partner volume or qtree ensures quick availability and restoration of data, from the point of the last Snapshot copy, should the storage system containing 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, thus freeing the original storage system of this time-consuming, performance-degrading chore. See Chapter 4, "Data Protection Using SnapMirror" on page 77.
SnapVault® (license required)	Centralized backup of multiple qtrees on multiple storage systems using Snapshot technology. This feature enables you to back up qtrees on multiple volumes and storage systems to a single SnapVault secondary storage system specialized for quick backup and restore of its sources. You can also install the Open Systems SnapVault agent on Windows NT, Windows 2000, Solaris, Linux, AIX, 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 storage system of this time-consuming, performance-degrading chore. See Chapter 5, "Data Protection Using SnapVault" on page 219.

Data protection feature	Description
Tape backup `dump` and `restore` commands	Tape backup and restore. The `dump` and `restore` commands allow you to back up Snapshot copies to tape. The `dump` command takes a Snapshot copy of the volume and then copies that data to tape. Because the Snapshot copy, not the active file system, is backed up to tape, Data ONTAP can continue its normal functions while the tape backup takes place. See the *Tape Backup and Recovery Guide* for more information.
`vol copy`	Fast block-copy of data from one volume to another. The `vol copy` command enables you to quickly block-copy stored data from one volume to another. See Chapter 7, "Volume Copy" on page 377.
SyncMirror® (cluster configuration required)	Continuous mirroring of data to two separate volumes. This feature allows you to mirror data real-time to matching volumes physically connected to the same storage system head. In case of unrecoverable disk error on one volume, the storage system automatically switches access to the mirrored volume. Cluster configuration is required for this feature. See Chapter 8, "SyncMirror Management" on page 401.
nvfail option to the `vol options` command	Protection against data corruption by failures of non-volatile RAM (NVRAM). See Chapter 9, "Database Protection Using NVFAIL" on page 441.
virus scan support	Support for third-party virus-scanning software for files accessed by CIFS clients. See Chapter 10, "Virus Protection for CIFS" on page 447.

Data protection feature	Description
MetroCluster	SyncMirror functionality enhanced to provide continuous volume mirroring over 500-meter to 30-kilometer distances.

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 the 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 SnapMirror feature allows you to schedule regular automatic copies of file system Snapshot copies of a volume or qtree onto another volume or qtree (on the same or a different storage system).
- The SnapRestore feature restores an entire volume to the state recorded in a previously created Snapshot copy with maximum speed and disk space efficiency.
- The SnapVault feature protects the data in one or more qtrees in a series of Snapshot copies stored on a separate storage system. SnapVault maintains an online, asynchronous, permanently read-only replica of the qtree data. SnapVault backup and Snapshot copy creation runs on an automated schedule.

  Note

  SnapVault, in addition to providing storage system backup, also provides direct backup to 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 storage system.
- The MetroCluster feature provides SyncMirror continuous mirroring over extended distances (500 meters to 30 kilometers).
- You can also use the `vol copy` command to snapshot and copy file systems to separate volumes or storage systems manually or by means of a script. You can use the `ndmpcopy` command to copy any subtree to any location on any storage system.

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. Storage systems 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 Snapshot copies to make a read-only image of a file system on the same disk	Chapter 2, "Snapshot Management" on page 17
Using SnapRestore to restore data to a corrupted volume from a previous Snapshot copy	Chapter 3, "Data Recovery Using SnapRestore" on page 61

Method	Chapter
Using SnapMirror to maintain a replica of one volume in another volume	Chapter 4, "Data Protection Using SnapMirror" on page 77
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 219
Using the `vol copy` command to copy data from one volume to another	Chapter 7, "Volume Copy" on page 377
Using SyncMirror to maintain two identical copies of a volume at all times	Chapter 8, "SyncMirror Management" on page 401

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 441 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 storage system.

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 storage system at a site

What a disaster affects

When a disaster occurs, it can affect all of your computing infrastructure including storage systems, 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 procedure to confirm that a disaster really has occurred. The procedure should determine the status of the supposed disaster site.

- Use external interfaces, 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 features and products are best for enabling the administrator to back up or replicate data stored on a local storage system 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 copy 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 copy 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 219.

Feature	Description
SnapMirror: Inter-site Snapshot copy backup, availability, and restorability	A SnapMirror destination storage device can be located offsite, any distance from the source system whose volumes it is mirroring. **Data availability:** In event of a data-loss disaster at a source site, SnapMirror data at 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 copy transfer from source to destination. **Connection requirements:** DSL connections or faster are recommended between the source and destination. Modem connections are possible. Routers, switches, and DNS servers should be pre-configured to direct users to the destination storage site if the source 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 77.

Feature	Description
Metro-Cluster: Inter-site real-time backup, availability, and restorability	Synchronously mirrored MetroCluster storage systems can be located at different sites, up to ten miles from one another. **Data availability:** In event of a data-loss disaster at one site, data that has been mirrored to the partner 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 system that they first attempt to access becomes unavailable. **Advantage:** Combined real-time off-site protection and availability.

Data protection in a SAN environment

Protecting volumes containing Data ONTAP LUNs

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

See your *Block Access Management Guide* for descriptions of data backup and restore on volumes containing Data ONTAP LUNs.

Snapshot Management 2

About this chapter

This chapter discusses what Snapshot™ copies are and how to manage them.

Topics in this chapter

This chapter discusses the following topics:

- "Understanding Snapshot copies" on page 18
- "Accessing Snapshot copies from clients" on page 20
- "Restoring files from Snapshot copies" on page 24
- "Creating Snapshot schedules" on page 25
- "Creating Snapshot copies" on page 32
- "Finding the Snapshot copy you need" on page 33
- "Understanding Snapshot disk consumption" on page 36
- "Saving disk space by using file folding" on page 48
- "Displaying data change rate between Snapshot copies" on page 49
- "Snapshot copy autodelete" on page 52
- "Displaying space reclaimed from deleted Snapshot copies" on page 55
- "Deleting Snapshot copies" on page 56
- "Renaming Snapshot copies" on page 59

Understanding Snapshot copies

What a Snapshot copy is

A Snapshot copy is a frozen, read-only image of a traditional volume, a FlexVol volume, or an aggregate that reflects the state of the file system at the time the Snapshot copy was created. Snapshot copies are your first line of defense for backing up and restoring data. See the *Storage Management Guide* for information about traditional volumes, FlexVol volumes, or aggregates.

When Snapshot copies are created

Data ONTAP maintains a configurableSnapshot schedule that creates and deletes Snapshot copies automatically for each volume. See "Creating Snapshot schedules" on page 25. Snapshot copies can also be created and deleted manually.

Maximum number of Snapshot copies

You can store up to 255 Snapshot copies at one time on each volume.

Maximum space Snapshot copies can occupy

You can specify the percentage of disk space that Snapshot copies can occupy. The default setting is 20% of the total (both used and unused) space on the disk. For a full explanation of how Snapshot copies consume disk space, see "Understanding Snapshot disk consumption" on page 36.

How Snapshot copies 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 (including permissions information) about files on the storage system. 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 copy 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 copy, 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 Snapshot copies

Snapshot copies enable system administrators to

- Create instantaneous backups
- Create a clone of a FlexVol volume
- Create a clone of a Data ONTAP LUN

Note

See the *Storage Management Guide* for information about cloning a FlexVol volume. See your *Block Access Management Guide* for information about cloning a Data ONTAP LUN.

Snapshot copies 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 Snapshot copies from clients

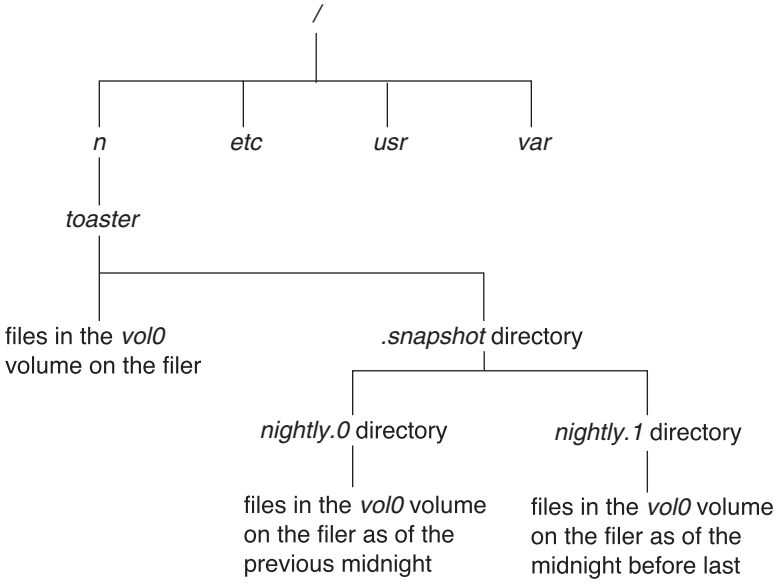
About user access to Snapshot copies

By default, every volume contains a directory named .snapshot, through which users can access old versions of files in that directory. How users gain access to Snapshot copies depends on the file-sharing protocol used—NFS or CIFS—and is described in the following sections. Access to Snapshot copies 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 volume can read that file in a Snapshot copy. A user without read permission to the volume cannot read that file in a Snapshot copy. Write permission does not apply because Snapshot copies are read-only. Snapshot copies can be accessed by any user with the appropriate permissions.

NFS user access to Snapshot copies

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



Explanation: In this example, the user can obtain access to Snapshot copies 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 file system, a directory listing looks like this:

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

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

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

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

```
systemA> 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 Snapshot copies 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 Snapshot copies

By default, CIFS users cannot see the .snapshot directory. Use the `cifs.show_snapshot` option to see the .snapshot directory. See the na_options(1) manual (man) page for details.

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 copy 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 Snapshot copies from CIFS clients

To access Snapshot copies 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: **`\\systemname\share\.snapshot`** (or **`~snapshot`** or **`~SNAPSHT`**). *systemname* is the name of the storage system you are using. *share* is the name of the share you want to access.

Example: `\\systemA\home\.snapshot`

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

Disabling and enabling client access to Snapshot copies

Sometimes you might want to disable and enable client access to Snapshot copies 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 Snapshot copies 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 copy access.
	Enable client access to Snapshot copies 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 copy access.

Restoring files from Snapshot copies

Why you restore files

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

Note

If you have purchased the SnapRestore option, you can automatically restore files or volumes from Snapshot copies with one command. To restore a volume, see "Deciding when to use SnapRestore" on page 63. To restore a single file, see "Reverting a file to a selected Snapshot copy" on page 70.

How to restore a file

To restore a file from a Snapshot copy, complete the following steps.

Step	Action
1	If the original file still exists and you do not want it overwritten by the snapshot file, then use your UNIX or Windows client to rename the original file or move it to a different directory.
2	Locate the Snapshot copy 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.

Snapshot restoration using Shadow Copy Client tools

You can access and restore Data ONTAP Snapshot files using the Windows Shadow Copy Client. The Shadow Copy Client provides a Previous Versions tab in the Properties menu from which you can view and restore Data ONTAP Snapshot images.

The Shadow Copy Client software for Windows 2003 is called the Previous Versions Client. Downloads available from Microsoft allow you to use Shadow Copy client tools on most older versions of Windows. Consult the Microsoft documentation for more information about Shadow Copy Client or Previous Versions Client software.

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 Snapshot copies automatically and deletes old Snapshot copies after a predetermined amount of time.

The default Snapshot schedule

When you install Data ONTAP on a storage system, it creates a default Snapshot schedule. The default Snapshot schedule automatically creates one nightly Snapshot copy Monday through Saturday at midnight, and four hourly Snapshot copies at 8 a.m., noon, 4 p.m., and 8 p.m. Data ONTAP retains the two most recent nightly Snapshot copies and the six most recent hourly Snapshot copies, and deletes the oldest nightly and hourly Snapshot copies when new Snapshot copies are created. You can see an example of the output for the default schedule in "Default snap schedule command results" on page 29.

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

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

Note

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

Snapshot schedule conflicts

If either the SnapMirror or SnapVault features are scheduled to perform Snapshot management at the same time as a `snap sched` activity, then the Snapshot management operations scheduled using the `snap sched` command might fail with syslog messages, "Skipping creation of hourly snapshot," and "Snapshot already exists."

To avoid this condition, stagger your Snapshot update schedules so that SnapMirror activity does not begin or end at the exact minute a `snap sched` operation attempts to create a Snapshot copy. Additionally, if `snap sched` Snapshot copies conflict with SnapVault activity, use the `snapvault snap sched` command to configure equivalent schedules.

Displaying the Snapshot schedule at the command line

To display the Snapshot schedule for a volume, 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 storage system.

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

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

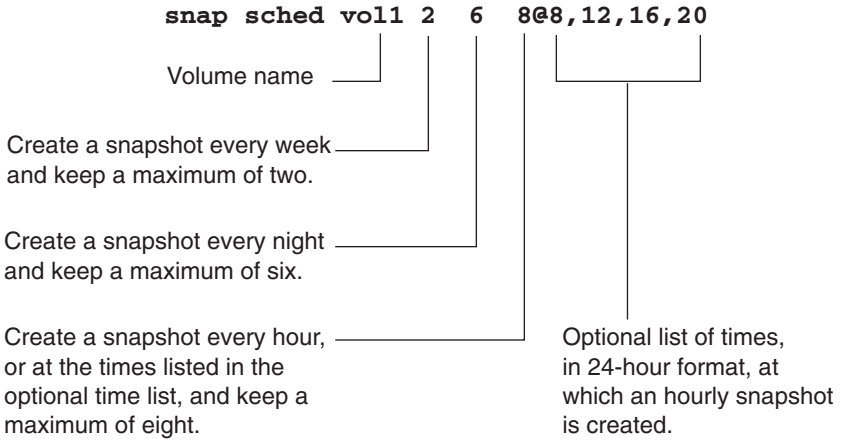
Displaying the Snapshot schedule using FilerView

To display the Snapshot schedule for a volume using FilerView, complete the following steps.

Step	Action
1	In FilerView, click Volumes in the list on the left.
2	In the list under Volumes, click Snapshots.
3	In the list under Snapshots, click Configure.

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 Snapshot copies, the six most recent nightly Snapshot copies, and the eight most recent hourly Snapshot copies, created at 8 a.m., noon, 4 p.m., and 8 p.m. every day. Whenever the Snapshot schedule creates a new Snapshot copy of

a particular type, it deletes the oldest one and renames the existing ones. On the hour, for example, the system deletes hourly.7, renames hourly.0 to hourly.1, and so on.

Note

If you omit the `@` argument specifying the hours for the hourly Snapshot copies, Data ONTAP creates a Snapshot copy every hour. Nightly and weekly Snapshot copies are always created at midnight.

Strategies for creating a Snapshot schedule

Following are some strategies for scheduling and retaining Snapshot copies:

- If users rarely lose files or typically notice lost files right away, use the default Snapshot schedule. This schedule creates no weekly Snapshot copy; it creates a Snapshot copy every night and keeps two; and it creates hourly Snapshot copies 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, delete the Snapshot copies less often than you would if you used the default schedule.

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

  ```
  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 Snapshot copies consumes 10% to 20% of disk space. Considering the benefits of Snapshot copies, it is worthwhile to reserve this amount of disk space for Snapshot copies. For more information on how Snapshot copies consume disk space, see "Understanding Snapshot disk consumption" on page 36.

- You can create different Snapshot schedules for different volumes on a storage system. On a very active volume, schedule Snapshot copies every hour and keep them for just a few hours, or turn off Snapshot copies. For example, the following schedule creates a Snapshot copy 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 Snapshot copies as long as they notice their mistake within a couple of hours.

- When you create a new volume, 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 Snapshot copies 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 copy. *weekly* is the number of weekly Snapshot copies to keep. *nightly* is the number of nightly Snapshot copies to keep. *hourly* is the number of hourly Snapshot copies to keep. *n,n,...* specifies the hours at which to create the hourly Snapshot copies. **Note** A zero in any of the three schedules (weekly, nightly, hourly) disables Snapshot copies 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 Snapshot copies 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 Snapshot copies are created at midnight of each day except Sunday, and weekly Snapshot copies are created at midnight on Sunday. Only one Snapshot copy is created at a time. If a weekly Snapshot copy is being created, for instance, no daily or hourly Snapshot copy will be created even if one is scheduled.

Disabling and enabling automatic Snapshot copies

You can disable automatic Snapshot copies for a period of time without changing the automatic Snapshot schedule.

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

Step	Action	
1	**If you want to...**	**Then...**
	Disable automatic Snapshot copies	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 Snapshot copies.
	Enable automatic Snapshot copies	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 Snapshot copies.

Note

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

snap sched *volume_name* 0 0 0

Creating Snapshot copies

Automatic or manual Snapshot copies

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

Creating Snapshot copies manually

To create a Snapshot copy 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 copy. *snapshot_name* is the name you want to give the Snapshot copy.

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.

Creating Snapshot copies manually using FilerView

To create Snapshot copies manually using FilerView, complete the following steps.

Step	Action
1	In FilerView, click Volumes in the list on the left.
2	In the list under Volumes, click Snapshots.
3	In the list under Snapshots, click Add.

Finding the Snapshot copy you need

Why you need to access a particular Snapshot copy

You might need to access a particular Snapshot copy (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 Snapshot copies 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 copy.

File version versus file access time

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

Finding all file versions in Snapshot copies

The best way to find all versions of a particular file preserved in Snapshot copies 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 Snapshot copies, 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 Snapshot copies 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 Snapshot copies, 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 copy, the access time of each file in the Snapshot copy is updated to the Snapshot copy creation time.

To determine when Snapshot copies 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 copy 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 copy, and are not related to the times at which the Snapshot copies 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 use information about the amount of disk space that Snapshot copies consume and are likely to consume to determine

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

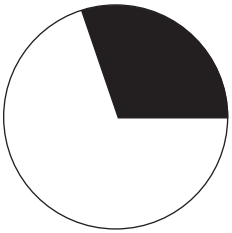
How Snapshot copies consume disk space

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

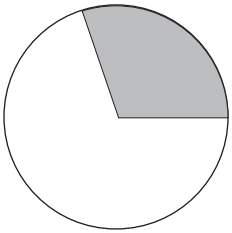
Disk consumption is minimized by preserving individual disk blocks rather than whole files. Snapshot copies begin to consume extra space only when files in the active file system are changed or deleted. When this happens, the original file blocks are still preserved as part of one or more Snapshot copies, 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 Snapshot copies to reflect what the active file system was before the change.

Example of how Snapshot copies consume disk space

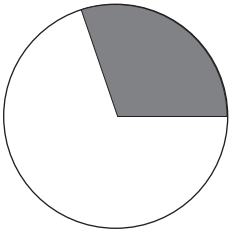
The following illustration shows how Snapshot copies consume disk space before and after you delete a file named myfile.txt.



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.

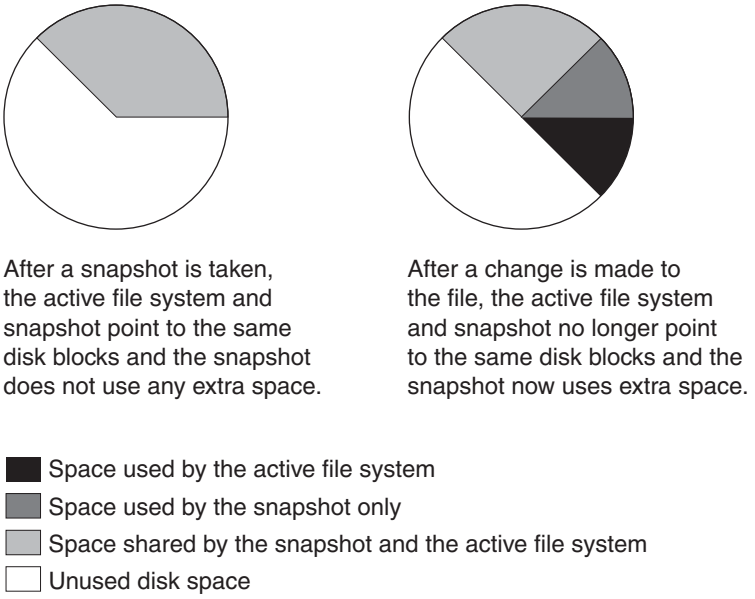


After *myfile.txt* is deleted from the active file system, the snapshot still includes the file and references its disk blocks. That's why deleting active file system data does not always free disk space.

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 copy 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 copy.



Disk consumption by multiple Snapshot copies

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

The same version of that 1-MB file might be referenced by several Snapshot copies: hourly.0, hourly.1, and hourly.2. If those Snapshot copies 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 Snapshot copies even though all three Snapshot copies 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 Snapshot copies and how to monitor Snapshot disk space usage:

- "Monitoring Snapshot disk consumption" on page 39
- "Displaying Snapshot disk consumption statistics" on page 40
- "Understanding the Snapshot reserve" on page 44
- "Changing the Snapshot reserve" on page 47

Monitoring Snapshot disk consumption

Displaying disk consumption

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

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

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

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

```
systemA> 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 Snapshot copies, 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 Snapshot copies 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 Snapshot copies 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 copy. 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 Snapshot copies in a volume

The `snap list` command shows the amount of disk space consumed by Snapshot copies in a specified volume. This command enables you to see how much disk space each Snapshot copy uses and helps you determine an appropriate Snapshot reserve. The `snap list` command also shows whether a Snapshot copy 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: **`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.

```
systemA> 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 Snapshot copies as a percentage of disk space being used in the volume. The first number is cumulative for all Snapshot copies listed so far, and the second number is for the specified Snapshot copy 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 Snapshot copies 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 copy and all other more recent Snapshot copies (the ones preceding this Snapshot copy 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 40 shows a volume that keeps five nightly Snapshot copies and four hourly Snapshot copies.

The sample output shows that the overhead for Snapshot copies 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 44.

The values in parentheses, which show the space used by an individual Snapshot copy, are useful in identifying a particular Snapshot copy to delete when the file system is full. However, deleting a particular Snapshot copy does not necessarily release the amount of disk space indicated, because other Snapshot copies 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 Snapshot copies to exceed a certain percentage of the used disk space, use the cumulative values in the `snap list` output to determine which Snapshot copies to delete. In the preceding example, if you do not want more than 5% of used disk space to be spent by Snapshot copies, delete all Snapshot copies listed below nightly.1 in the `snap list` output; that is, nightly.2, nightly.3, and nightly.4. After deleting the Snapshot copies, nightly.1 and all the other more recent Snapshot copies consume 5% of the used disk space.

Displaying Snapshot copy use and dependencies

The `snap list` command displays the notation "busy" after the name of a Snapshot copy if the Snapshot copy is being actively used by an application, as shown the example below for the Snapshot copy hourly.0. Busy Snapshot copies 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 copy name if the application needs the Snapshot copy currently or at a later time. For example, "backup" is displayed next to the Snapshot copy name to

show that the Snapshot copy 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 copy name means that SnapMirror is retaining the Snapshot copy to maintain a source-destination relationship.

Note

Ownership information for a busy Snapshot copy is useful for determining whether to stop the activity in progress. For example, if the `snap list` command output displays a locked Snapshot copy that is imposing a resource constraint, you can delete that Snapshot copy and free up space.

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 can take up to several minutes to complete. This condition persists until the Snapshot copy from which you 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 copy" on page 70.

Understanding the Snapshot reserve

What the Snapshot reserve is

The Snapshot reserve specifies a set percentage of disk space for Snapshot copies. By default, the Snapshot reserve is 20% of disk space. The Snapshot reserve can be used only by Snapshot copies, 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 Snapshot copies, Snapshot copies 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 Snapshot copies so that they do not consume active file system space
- Keeping disk space consumed by Snapshot copies 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 Snapshot copies in the Snapshot reserve, deleting files in the active file system frees disk space for new files, while the Snapshot copies that reference those files consume only the space in the Snapshot reserve. If Data ONTAP created a Snapshot copy 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 copy. Data ONTAP would have to delete the Snapshot copy 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 copy:

If Data ONTAP creates a Snapshot copy 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 Snapshot copies 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 Snapshot copies, 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%
```

Snapshot copies can exceed reserve

There is no way to prevent Snapshot copies from consuming disk space greater than the amount reserved for them; therefore, it is important to reserve enough disk space for Snapshot copies 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 Snapshot copies, along with the 500,000 KB (0.5 GB) that were being used by Snapshot copies 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 Snapshot copies; 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 Snapshot copies 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 Snapshot copies.

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 copy, every disk block in the file system is referenced by some Snapshot copy. 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 Snapshot copies to free more disk space.

See "Displaying Snapshot disk consumption statistics" on page 40 for information about how to use the `snap list` command to determine which Snapshot copies to delete.

Changing the Snapshot reserve

Changing the Snapshot reserve

After you understand how Snapshot copies 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. *percent* is the percentage of disk space you want to reserve for Snapshot copies.

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 copy, and, if it is identical to the Snapshot copy currently being created, just referencing the previous Snapshot copy instead of taking up disk space writing the same data in the new Snapshot copy. 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 copy, if any.

The system 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.

Displaying data change rate between Snapshot copies

What is displayed

You can display the rate of change stored between two Snapshot copies as well as the rate of change between a Snapshot copy and the active file system. Data ONTAP displays the rates of change in two tables. The first table displays rates of change between successive Snapshot copies. The second table displays a summary of the rate of change between the oldest Snapshot copy and the active file system. See the na_snap(1) man page for details.

Displaying rates of change on a volume

To display data change rates on a volume, complete the following step.

Step	Action
1	Enter the following command: **`snap delta vol_name`** *vol_name* is the name of the volume containing the Snapshot copies. **Note** You can display change rates for all volumes by omitting the volume name

Example: The following command lists the rates of change for the vol0 volume:

```
system> snap delta vol0
Volume vol0 working...
From Snapshot To                 KB changed Time       Rate (KB/hour)
------------- ------------------ ---------- ---------- --------------
hourly.0      Active File System 149812       0d 03:43 40223.985
hourly.1      hourly.0           326232       0d 08:00 40779.000
hourly.2      hourly.1           2336         1d 12:00 64.888
hourly.3      hourly.2           1536         0d 04:00 384.000
hourly.4      hourly.3           1420         0d 04:00 355.000
nightly.0     hourly.4           1568         0d 12:00 130.666
hourly.5      nightly.0          1400         0d 04:00 350.000
nightly.1     hourly.5           10800      201d 21:00 2.229
Summary...
From Snapshot To                 KB changed Time       Rate (KB/hour)
------------- ------------------ ---------- ---------- --------------
nightly.1     Active File System 495104     204d 20:43 100.697
```

Displaying rates of change between Snapshot copies

To display data change rates between two Snapshot copies, complete the following step.

Step	Action
1	Enter the following command: **`snap delta vol_name snap1 snap2`** *vol_name* is the name of the volume containing the Snapshot copies. *snap1* and *snap2* are the names of the two Snapshot copies.

Example: The following command lists the rate of change between nightly.0 and hourly.1 of the vol0 volume:

```
system> snap delta vol0 nightly.0 hourly.1
Volume vol0 working...
From Snapshot To                 KB changed Time       Rate (KB/hour)
------------- ------------------ ---------- ---------- --------------
hourly.2      hourly.1           2336         1d 12:00 64.888
hourly.3      hourly.2           1536         0d 04:00 384.000
hourly.4      hourly.3           1420         0d 04:00 355.000
nightly.0     hourly.4           1568         0d 12:00 130.666
Summary...
From Snapshot To                 KB changed Time       Rate (KB/hour)
------------- ------------------ ---------- ---------- --------------
nightly.0     hourly.1           6860         2d 08:00 122.500
```

Snapshot copy autodelete

Defining a policy for automatically deleting Snapshot copies

To define and enable a policy for automatically deleting Snapshot copies, complete the following step:

Step	Action	
1	Enter the following command: **`snap autodelete vol-name on option value`** on enables the Snapshot copy autodelete policy. You can define which Snapshot copies you delete by entering the following options and their values:	
	Option	**Values**
	commitment	Specifies whether a Snapshot copy is linked to data protection utilities (SnapMirror or NDMPcopy) or data backing mechanisms, (volume or LUN clones). ◆ try—delete only Snapshot copies that are not linked to data protection utilties and data backing mechanisms. ◆ disrupt—delete only Snapshot copies that are not linked to data backing mechanisms.
	trigger	Defines when to begin automatically deleting Snapshot copies. ◆ volume—begin deleting Snapshot copies when the volume is nearly full ◆ snap_reserve—begin deleting Snapshot copies when the Snapshot reserve is nearly fully. ◆ space_reserve—begin deleting Snapshot copies when the space reserved in the volume is nearly full.
	target_free_space	Determines when to stop deleting Snapshot copies. Specify a percentage. For example, if you specify 30, then Snapshot copies are deleted until 30 percent of the volume is free space.

Step	Action	
	delete_order	◆ newest_first—delete the most recent Snapshot copies first. ◆ oldest_first—delete the oldest Snapshot copies first.
	defer_delete	Delete one of the following types of Snapshot copies last: ◆ user_created—Snapshot copies that are not autoscheduled ◆ prefix—Snapshot copies with the specified prefix_string
	prefix	Delete Snapshot copies with a specific prefix last. You can specify up to 15 characters (for example, sv_snap_week). Use this option only if you specify prefix for the defer_delete option.

Viewing current Snapshot copy autodelete settings

To view current autodelete settings, complete the following step:

Step	Action
1	Enter the following command: **`snap autodelete`** ***`vol-name`*** **`show`** **Result:** Snapshot copy autodelete settings revert to the following defaults: ◆ state—off ◆ commitment —try ◆ trigger—volume ◆ target_free_space— 20% ◆ delete_order—oldest_first ◆ defer_delete—user_created ◆ prefix— no prefix specified

Restoring default Snapshot copy autodelete settings

To restore default Snapshot copy autodelete settings, complete the following step.

Step	Action
1	Enter the following command: **`snap autodelete vol-name reset`** *vol-name* is the name of the volume. **Result:** Snapshot copy autodelete settings revert to the following defaults: ◆ state—off ◆ commitment —try ◆ trigger—volume ◆ target_free_space— 20% ◆ delete_order—oldest_first ◆ defer_delete—user_created ◆ prefix— no prefix specified

Disabling a Snapshot copy autodelete policy

To disable a Snapshot copy autodelete policy complete the following step.

Step	Action
1	Enter the following command: **`snap autodelete vol-name off`** *vol-name* is the name of the volume. **Result:** Snapshot copies are not automatically deleted when the volume is nearly full.

Displaying space reclaimed from deleted Snapshot copies

About reclaimed space

You can display the amount of space you can reclaim by deleting one or more Snapshot copies in a volume. The amount of space displayed is an approximation because writing to the volume, creating Snapshot copies, or deleting Snapshot copies cause the reclaimed amount to change.

Displaying amount of reclaimed space

To display the amount of space you can reclaim by deleting Snapshot copies, complete the following step.

Step	Action
1	Enter the following command: **`snap reclaimable vol_name snap1 [snap2 ...]`** *vol_name* is the volume which contains the Snapshot copies you might delete. *snap1* [*snap2* ...] are the names of Snapshot copies you might delete. The names are separated by a space. **Note** It might take a while for Data ONTAP to display the amount of freed space. You can press Ctrl-C to interrupt the command.

Example: The following command displays the amount of space reclaimed by deleting the hourly.4, hourly.5, and nightly.0 Snapshot copies in the vol1 volume:

```
system> snap reclaimable vol1 hourly.4 hourly.5 nightly.0
Processing (Press Ctrl-C to exit) ...
snap reclaimable: Approximately 240 kbytes would be freed.
```

Deleting Snapshot copies

How to determine which Snapshot copies to delete on the basis of size

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

In "Sample snap list command output" on page 40, the cumulative disk space used by Snapshot copies 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 Snapshot copies is the same. In this case, deleting only one of the Snapshot copies 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, 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 Snapshot copies are consuming in the volume. If the amount seems high, decrease the amount of time that Snapshot copies are kept or increase the Snapshot reserve.

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

Caution

As a general rule, avoid deleting Snapshot copies that are not the product of the `snap sched` command (for example, Snapshot copies generated by SnapMirror or SnapVault commands). Deleting these Snapshot copies could halt the SnapMirror or SnapVault processes. An exception would be Snapshot copies left over from old SnapMirror relationships that you no longer want to maintain. See "Releasing partners from a SnapMirror relationship" on page 171.

Deleting a Snapshot copy manually

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

To delete a Snapshot copy 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 copy to delete. *snapshot_name* is the specific Snapshot copy to delete. **Note** To delete all Snapshot copies on a volume, use the `-a` parameter: **`snap delete -a`** *volume_name*

Deleting a Snapshot copy manually if the Snapshot copy is busy

The `snap delete` command displays ownership information for busy Snapshot copies. This information is useful for determining why a particular Snapshot copy is busy, and whether to stop the activity in progress. For example, if the `snap delete` command output displays a locked Snapshot copy that is imposing a resource constraint, you can delete that Snapshot copy and free up space.

Deleting a Snapshot copy manually if the Snapshot copy is locked

If a Snapshot copy 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 171) or `snapvault release` (see "Releasing SnapVault relationships" on page 297) command to unlock the Snapshot copy. Snapshot copies are locked because SnapMirror or SnapVault is maintaining them for the next update. Deleting a locked Snapshot copy 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 copy

The following example shows how you would delete a SnapMirror Snapshot copy that was locked because SnapMirror needed it for an update:

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

Example of how to delete a locked SnapVault Snapshot copy

The following example shows how you would delete a SnapVault Snapshot copy that was locked because SnapVault needed it for an update:

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

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

systemA> snapvault release /vol/vol0/qt3 systemB:/vol/sv_vol/qt3

systemA> snap delete vol0 oldsnap
```

Renaming Snapshot copies

Why to rename a Snapshot copy

You might want to rename a snapshot generated by the `snap sched` command 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 Snapshot copies automatically, by using the `snap rename` command.

How to rename a Snapshot copy

To rename a Snapshot copy, 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 copy to rename. *from_name* is the current name of the Snapshot copy to rename. *to_name* is the new name you want to give to the Snapshot copy.

Example: `snap rename vol0 hourly.2 MyDataSave`

Data Recovery 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 62
- "Deciding when to use SnapRestore" on page 63
- "Installing the SnapRestore license" on page 66
- "Reverting a volume to a selected Snapshot copy" on page 67
- "Reverting a file to a selected Snapshot copy" on page 70
- "Obtaining correct incremental backups after reversion" on page 75

Understanding SnapRestore

What SnapRestore does

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

How SnapRestore works

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

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 copy taken before the data corruption.

Why use SnapRestore rather than copying from a Snapshot copy

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 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 copy.

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 copy 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 system reboots with configuration files that were in effect when the Snapshot copy 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 copy 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 copy 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 copy 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 copy from a previous release of Data ONTAP. However, doing so can cause problems because of potential version incompatibilities and can prevent the system from booting completely.

Prerequisites

You must meet these prerequisites before using SnapRestore:

- SnapRestore must be licensed on your storage system.
- There must be at least one Snapshot copy on the system that you can select to revert.
- 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 copy, you cannot undo the reversion.
- If you revert to a Snapshot copy taken before a SnapMirror Snapshot copy, Data ONTAP can no longer perform an incremental update of the mirror; it must re-create the baseline version of the mirror.
- Snapshot copy deletions are irrevocable. If you delete a Snapshot copy, you cannot recover the Snapshot copy by using SnapRestore.
- After you revert a volume to a selected Snapshot copy, you lose all the Snapshot copies that were taken after the selected Snapshot copy.
- Between the time you enter the `snap restore` command and the time when reversion is completed, Data ONTAP stops deleting and creating Snapshot copies.
- If you are reverting a file from a Snapshot copy, you can delete other Snapshot copies, except for the Snapshot copy you are reverting from.

Caution about reverting the root volume

Because the /etc directory of the root volume contains configuration information about the system, reverting the root volume might change the configuration. In addition, restoring the root volume restores options settings for the entire system to the settings that were in effect when the Snapshot copy 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 storage system before using SnapRestore. After you revert the root volume, restore the /etc directory and reboot. If you back up the /etc directory to another volume, you can use the following command to make the system reboot with that volume as the root volume: `vol options` *`volume`* `root` In this way, when the system reboots during a revert, 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 copy

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 copy. **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 copy for reverting each volume you want to revert	Go to Step 6.
	You want to choose a Snapshot copy from the list of Snapshot copies available for reversion	Go to Step 3.

Step	Action
3	Enter the following command: **`snap restore -t vol volume_name`** `-t vol` 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. **Note** Use the -f option to avoid warning messages and prompts to confirm your decision to revert the volume. See the na_snap(1) man page for more information.
4	Press **`y`** to confirm that you want to revert the volume. **Result:** Data ONTAP displays a list of Snapshot copies.
5	Enter the name of the Snapshot copy 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 copy to be used for the reversion.
6	Enter the following command: **`snap restore -t vol -s snapshot_name volume_name`** `-t vol` specifies the volume name to revert. `-s` *snapshot_name* specifies the name of the Snapshot copy from which to revert the data. You can enter only one Snapshot copy 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.
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.

Step	Action	
8	**If...**	**Then...**
	You want to continue with the reversion	Press **y**. **Result:** The system reverts the volume from the selected Snapshot copy. If you are reverting the root volume, the system reboots.
	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 prompt.

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

system> 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 copy.

Reverting a file to a selected Snapshot copy

About snap restore file reversion

Using `snap restore` to revert a single file to a selected Snapshot copy is practical when the file is so large that you cannot copy the previous file version from the Snapshot copy 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 take up to several minutes to complete. You can minimize the amount of time required to complete by using the `snap list -n` command. See the manual (man) pages for more details.

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 copy. **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 copy 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 copy from the list of Snapshot copies 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 -t file -r restore_as_new_path path_and_file_name</tt>

<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 copy. 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.

Note
Use the <tt>-f</tt> option to avoid warning messages and prompts to confirm your decision to revert the volume. See the na_snap(1) man page for more information.

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 Snapshot copies.</td></tr>
</table>

Step	Action
5	Enter the name of the Snapshot copy 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 copy to be used for the reversion.
6	Enter the following command: **`snap restore -t file -s snapshot_name -r restore_as_path path_and_file_name`** `-t file` specifies that you are entering the name of a file to revert. `-s` *snapshot_name* specifies the name of the Snapshot copy from which to revert the data. `-r` *restore_as_path* restores the file to a location different from the location in the Snapshot copy. 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.
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 copy for the reversion and, if you have not used the `-f` option, prompts you to decide whether to proceed with the reversion.

Step	Action	
8	**If...**	**Then...**
	You want to continue with the reversion	Press **y**. **Result:** The storage system reverts the file from the selected Snapshot copy.
	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 prompt.

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

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

system> 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 copy.

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 copy 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 79
- "Effects of FlexVol volumes on SnapMirror replication" on page 90
- "Synchronous SnapMirror" on page 93
- "SnapMirror commands and configuration files" on page 96
- "Considerations when planning and running SnapMirror" on page 100
- "Setting up a basic SnapMirror operation" on page 105
- "Specifying destination systems on the source" on page 113
- "Defining source and destination through snapmirror.conf" on page 116
- "Using SnapMirror over multiple paths" on page 126
- "Enabling SnapMirror" on page 128
- "Initializing a SnapMirror destination" on page 130
- "Updating a destination manually" on page 138
- "Listing SnapMirror Snapshot copies" on page 141
- "Checking SnapMirror data transfer status" on page 144
- "Checking SnapMirror data transfer logs" on page 153
- "Aborting a SnapMirror transfer" on page 158
- "Stabilizing (quiescing) destinations before a Snapshot copy" on page 160
- "Turning off or changing scheduled updates for volumes or qtrees" on page 163
- "Turning off SnapMirror updates" on page 166
- "Converting a destination to a writable volume or qtree" on page 168
- "Releasing partners from a SnapMirror relationship" on page 171
- "Resynchronizing SnapMirror" on page 174
- "Migrating data between volumes by using SnapMirror" on page 182
- "Moving volume SnapMirror or qtree SnapMirror sources" on page 184

- "Copying from one destination to another in a series (cascading)" on page 190
- "Using SnapMirror to copy a volume to local tape" on page 196
- "How SnapMirror works with the dump command" on page 208
- "Fixing changes to SnapMirror elements" on page 209
- "Creating SnapLock destination volumes" on page 211
- "Protecting SnapVault secondaries using volume SnapMirror" on page 216

SnapMirror overview

SnapMirror modes

The Data ONTAP SnapMirror feature enables an administrator to mirror Snapshot images either asynchronously or synchronously:

- When mirroring asynchronously, SnapMirror replicates 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.
- When mirroring synchronously, SnapMirror replicates Snapshot images from a source volume to a partner destination volume at the same time it is written to the source volume.

  Additionally, you can configure a synchronous SnapMirror replication to lag behind the source volume by a user-defined number of write operations or milliseconds. This option is useful if you are balancing the need for synchronous mirroring with the performance benefit of asynchronous mirroring.

  See "Synchronous SnapMirror" on page 93 for more information.

SnapMirror can also be used with traditional volumes and FlexVol volumes. See "Effects of FlexVol volumes on SnapMirror replication" on page 90 for more information about using SnapMirror with FlexVol volumes.

Reasons for accessing destination volume information

You can access the information on the destination volume or qtree to

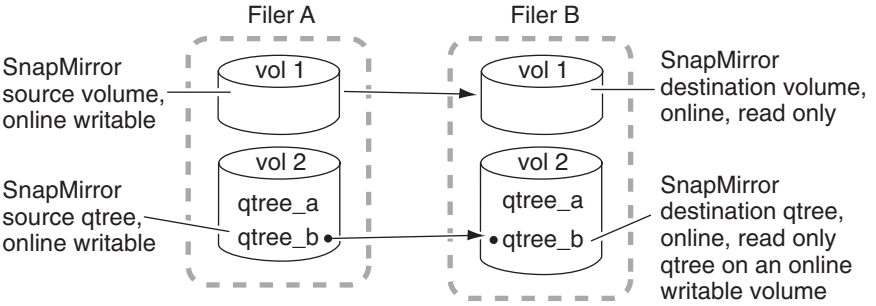
- Provide users quick access to mirrored data in the event of a disaster that makes the source volume or qtree unavailable
- Update the source to recover from disaster, data corruption (qtrees only), or user error
- Archive the data to tape
- Balance resource loads
- Back up or distribute the data to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

The components of SnapMirror

The basic SnapMirror deployment consists of the following components.

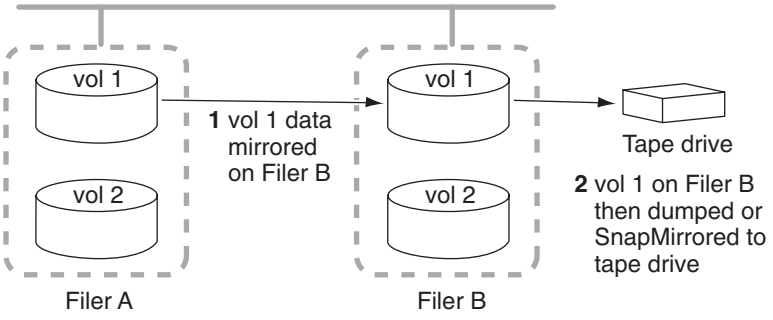
Source volumes or qtrees: SnapMirror source volumes and qtrees are writable data objects whose data is to be replicated. The source volumes and qtrees are the objects normally visible, accessible, and writable by the storage system clients.

Destination volumes or qtrees: The SnapMirror destination volumes and qtrees are read-only objects, usually on a separate system, 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.



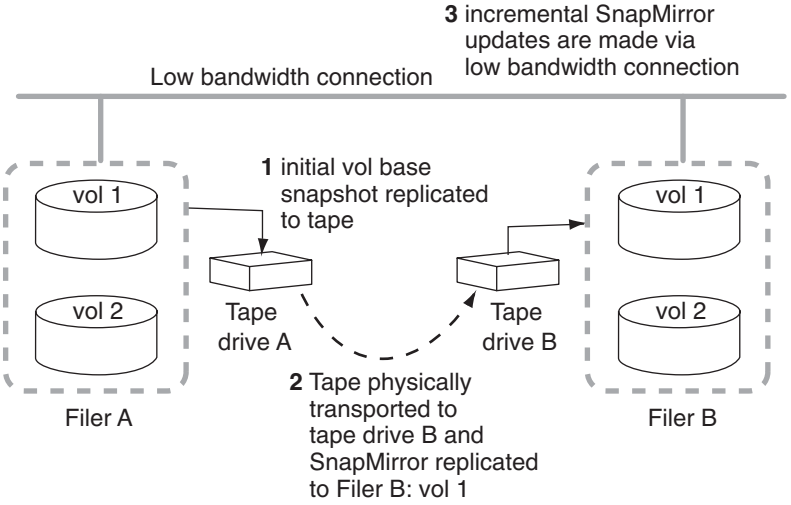
Source to destination to tape variation

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.

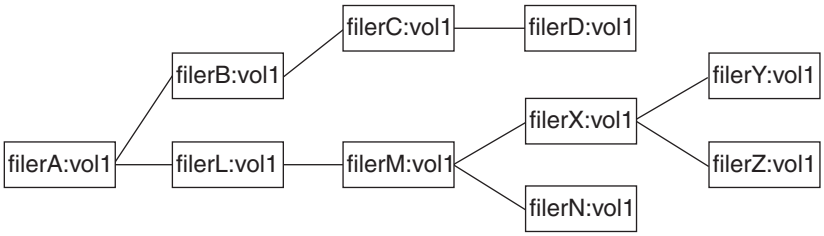


Cascading destinations variation

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. See “Copying from one destination to another in a series (cascading)” on page 190 for more information.

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 copy 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 379. 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 system 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.

Situation	How to use SnapMirror
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.
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 storage systems to distribute the load.
Off-loading tape backups: You need to reserve all processing and networking resources on a storage system for serving NFS and CIFS requests.	After copying data on the source system, you can back up the data in the destination to tape. This means that the source system 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 storage systems that are geographically closer to the users. Users accessing a local system can read the data using less resource time than if they connected to a distant system.

Using SnapMirror with SnapDrive™ software

If you are using SnapDrive software, use SnapMirror to replicate your data. See the *SnapDrive Installation and Administration Guide* for your version of SnapDrive for more information.

Note

SnapDrive supports the use of volume SnapMirror only. It does not support qtree SnapMirror replication.

Using SnapMirror with MultiStore® software

If you are using MultiStore software, use SnapMirror to migrate or create a disaster recovery vFiler™ unit. See the *MultiStore Management Guide* for more information.

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 Snapshot copies of data from a source volume to a destination volume, as long as the volumes are of the same type. When using the `vol copy` and `snapmirror` commands, the source and destination volumes must both be traditional volumes or FlexVol volumes. However, these commands 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 volumes.
- The SnapMirror feature supports incremental Snapshot updates between source and destination 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 or individual qtrees on a volume.

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

- SnapMirror volume replication can be synchronous or asynchronous.
- A destination volume is read-only.
- SnapMirror volume replication can only occur with volumes of the same type, that is, both traditional volumes or both FlexVol volumes.
- SnapMirror volume replication copies a Snapshot copy 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 volume replication is a block-for-block replication; it transfers the file system verbatim. Therefore, older releases of Data ONTAP cannot understand file system transfers from a later release of Data ONTAP.

Note

See the caution in "Prerequisites to running SnapMirror" on page 100 for information about source and destination upgrading order.

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

- SnapMirror qtree replication is asynchronous only.
- SnapMirror qtree replication occurs between qtrees regardless of the type of volume, traditional or flexible, in which the qtree resides.
- 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.
- SnapMirror qtree replication is a logical replication; all of the files and directories in the source file system are created in the destination file system. Therefore, replication can occur from any release to any release.

Note

If the source file system contains a file type that cannot be represented on the destination file system, the replication will fail. For example, Data ONTAP 7.0 supports files up to 16 TB in size, whereas earlier versions of Data ONTAP support files up to 4 TB. If the source system is running Data ONTAP 7.0, the qtree you want to replicate contains a file greater than 4 TB, and the destination system is running an earlier version of Data ONTAP, the replication will fail.

Differences between qtree SnapMirror and SnapVault

The following are differences between qtree SnapMirror and SnapVault:

- Qtree SnapMirror is more likely to be used in an environment requiring an immediate failover capability.
- SnapVault is likely to be used with applications that can afford to lose some data (backup) and not require immediate failover.
- Due to its heterogeneous capabilities, SnapVault is more likely to be used to protect remote offices.
- Qtree SnapMirror does not allow Snapshot creation or deletion on the read-only destination unless the relationship is broken and the destination is brought online. SnapVault adds Snapshot scheduling, retention, and expiration, providing versions (backups) on the destination.

- In certain situations, qtree SnapMirror allows replication in both directions because a SnapMirror source and SnapMirror destination can run on the same storage system. SnapVault replicates in one direction; a source and destination cannot run on the same storage system.
- Qtree SnapMirror software uses the same software and licensing on the source and destination storage systems. SnapVault software has a SnapVault source system and a SnapVault destination system, which provide different functionality.
- SnapMirror transfers can be scheduled as frequently as once every minute. SnapVault transfers can be scheduled once every hour.
- Multiple qtrees within the same source volume use one Snapshot copy each on the source system when qtree SnapMirror is used. Only one Snapshot copy total is used when SnapVault software is used.
- Qtree SnapMirror Snapshot copies are deleted by qtree SnapMirror software when they are no longer needed for replication purposes. SnapVault Snapshot copies are retained and deleted on a specified schedule.
- Qtree SnapMirror relationships can be reversed, allowing the source to be resynchronized with changes made at the destination. SnapVault provides the capability to transfer data from the destination to the source only for restore purposes. The direction of replication cannot be reversed.
- Qtree SnapMirror can be used to replicate data between storage systems running Data ONTAP only. SnapVault can be used to back up both NetApp and open systems sources, although the destination storage system must be a NetApp storage system.

Maximum number of simultaneous replication operations

A volume SnapMirror, qtree SnapMirror, or SnapVault replication consists of two operations, one operation being on the source side of the transfer and the other operation being on the destination side of the transfer. Therefore, if a storage system is the source of one replication and the destination of another replication, it uses two replication operations. Likewise, if a storage system is the source and destination of the same replication, it uses two replication operations.

The maximum number of simultaneous replication operations that each storage system model can support is shown in the following table.

Model	Number of simultaneous operations allowed
F85	4
F87	4

Model	Number of simultaneous operations allowed
F720	4
F740	4
F760	4
F810	8
F820	8
F825	8
F840	16
F880	16
FAS250	4
FAS270	8
FAS920	8
FAS940	16
FAS960	16
FAS980	16
FAS3020	16
FAS3050	16
GF270	8
GF825	8
GF940	16
GF960	16
GF980	16
R100	128
R150	128
R200	128

Storage systems that use ATA drives: Storage systems that have any ATA drives, other than NearStore® systems, have half the maximum number of simultaneous replications that the same storage system has using FC drives. The following table lists the maximum number of simultaneous replication operations for storage systems that can use FC drives and ATA drives.

Model	Maximum number of simultaneous operations allowed (FC drives only)	Maximum number of simultaneous operations allowed (if ATA drives are present)
FAS920	8	4
FAS940	16	8
FAS960	16	8
FAS3020	16	8
FAS3050	16	8

Factors that might reduce the maximum number of operations: A storage system might not reach the maximum number of simultaneous replication operations for the following reasons:

- Filer resources, such as CPU usage, memory, disk bandwidth, or network bandwidth, are taken away from SnapMirror or SnapVault operations.
- Each storage system in a cluster has the maximum number of simultaneous replication operations listed in the preceeding tables. If a failover occurs, the surviving storage system cannot process more than the maximum number of simultaneous replication operations specified for that storage system. These operations can be ones that were scheduled for the surviving storage system, the failed over storage system, or both. For example, each FAS960 in a cluster can run a maximum of 16 simultaneous replication operations. If one FAS960 fails over to the other, the surviving FAS960 still has a maximum of 16 operations, which can be ones that were scheduled by the surviving FAS960, the failed FAS960, or both.

  Note

  When planning SnapMirror or SnapVault replications using clusters, take this limitation into consideration.

- NearStore systems are optimized as a destination for qtree SnapMirror and SnapVault replication operations. Therefore, the number of simultaneous replication operations shown in the table represents the total number of

simultaneous qtree SnapMirror or SnapVault replication operations for which the filer is the destination. Replication operations of which the NearStore system is the qtree SnapMirror source, SnapVault source, volume SnapMirror source, or volume SnapMirror destination count twice against the maximum number.

Example: You have a NearStore system that is the destination of 20 qtree SnapMirror relationships and the source of 5 volume SnapMirror relationships to another storage system. If all the replications occur simultaneously, the NearStore system will have 30 replication operations running concurrently.

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 once per minute until it succeeds, SnapMirror is turned off, or the update is terminated.

Effects of FlexVol volumes on SnapMirror replication

Supported SnapMirror configurations

FlexVol volumes have enhanced SnapMirror replication, but they have also added to the number of SnapMirror configurations possible. The following sections describe supported configurations and note possible exceptions.

Volume SnapMirror: The following volume SnapMirror configurations, both synchronous and asynchronous, are supported:

- Traditional volume to traditional volume
- FlexVol volume to FlexVol volume

  There are two exceptions in which synchronous FlexVol to FlexVol SnapMirror replications are not supported:

  - A storage system cannot be both the source of a synchronous SnapMirror relationship using FlexVol volumes and the destination of another synchronous SnapMirror relationship using FlexVol volumes.
  - You cannot have two clustered systems that have bidirectional synchronous SnapMirror relationships between them. For example, if Cluster A and Cluster B are two different clustered storage systems, you cannot have some synchronous SnapMirror relationships going from Cluster A to Cluster B and some synchronous SnapMirror relationships going from Cluster B to Cluster A.

Note

Volume SnapMirror replication between different volume types is not supported. Specifically, traditional volume to FlexVol volume replication and FlexVol volume to traditional volume replication are not supported.

Qtree SnapMirror: The following qtree SnapMirror configurations are supported:

- Traditional volume to traditional volume
- Traditional volume to FlexVol volume
- FlexVol volume to traditional volume
- FlexVol volume to FlexVol volume

Migration from traditional volumes to FlexVol volumes

Only qtree SnapMirror can be used to migrate data from traditional volumes to FlexVol volumes, if you use SnapMirror for this type of migration. Volume SnapMirror cannot be used because it cannot replicate to a different type of volume. See "Differences between SnapMirror volume replication and qtree replication" on page 84.

Resynchronization of FlexVol volumes

Only qtree SnapMirror can be used to resynchronize data, using the `snapmirror resync` command, after the destination is used and the source needs to match. Volume SnapMirror cannot be used because resynchronization between newer versions of Data ONTAP that contain FlexVol volumes as a feature and older versions of Data ONTAP are not allowed. See "Differences between SnapMirror volume replication and qtree replication" on page 84.

No symmetrical disk geometry issue

FlexVol volumes make disk geometry less of an issue (see "Recommendations" on page 102). Destination volumes do not have to contain the same number of disks or the same size disks as the source volumes.

FlexClone impact on SnapMirror

FlexClone™ volumes create a nearly instantaneous replica of a volume within the same aggregate. For information about FlexClone volumes, see the FlexVol volume management section of the *Storage Management Guide*.

Cloning SnapMirror volumes: You can clone SnapMirror volumes. Note that a clone of a SnapMirror destination locks down the Snapshot copy from which the clone was created, in addition to locking down that Snapshot copy in the source volume and every volume in the cascade (if the volume is part of a SnapMirror cascade).

Note

A FlexClone volume can be created from a Snapshot copy in a SnapMirror destination, but a FlexClone volume cannot be the destination of a SnapMirror relationship.

Cloning from a Snapshot copy that is not the most recent Snapshot copy: If a FlexClone volume is created from a Snapshot copy that is not the most recent Snapshot copy, and that Snapshot copy no longer exists on the source volume, then every update will need to delete the Snapshot copy on the destination. In this case, all SnapMirror updates to the destination volume fail until the clone is destroyed or split.

Note

This will not occur if the clone is created from the most recent Snapshot copy in the SnapMirror destination because that Snapshot copy is guaranteed to exist in the source volume.

Effects of deleting the source Snapshot copy: If a source Snapshot copy for a SnapMirror relationship is deleted, SnapMirror fails on subsequent update requests. The failures occur because FlexClone volumes are bound to the source Snapshot copy regardless of whether the FlexClone volume was created on the source or destination volume. Even if the FlexCone volume was created remotely on the destination system, deleting the source Snapshot copy on the source system results in a failed SnapMirror update when attempting to propagate the deletion to the destination.

Synchronous SnapMirror

What synchronous SnapMirror is

Synchronous SnapMirror is a SnapMirror feature in which the data on one system is replicated on another system at, or near, the same time it is written to the first system. Synchronous SnapMirror synchronously replicates data between single or clustered storage systems situated at remote sites using either an IP or a Fibre Channel connection.

Note

You can use Synchronous SnapMirror only on volumes, not qtrees.

How SnapMirror replicates data synchronously

Before Data ONTAP saves data to disk, it collects written data in NVRAM. Then, at a point in time called a consistency point, it sends the data to disk. When the Synchronous SnapMirror feature is enabled, the source system forwards data to the destination system as it is written in NVRAM. Then, at the consistency point, the source system sends its data to disk and tells the destination system to also send its data to disk. Finally, the source system waits for the destination system to acknowledge that it sent data to disk before continuing with the next write.

How network problems are handled

If there are problems with your network, your synchronous replication might go into an asynchronous mode. Ordinarily, the source and destination systems periodically communicate with each other. In the event of a network outage, synchronous SnapMirror goes into an asynchronous mode if the periodic communication is disrupted. When in asynchronous mode, the source system tries to communicate with the destination system once every minute until communication is reestablished. Once reestablished, the source system asynchronously replicates data to the destination every minute until a synchronous replication can be reestablished.

Considerations before running synchronous SnapMirror

You should consider the following criteria before using the synchronous SnapMirror feature:

Not all storage systems are supported: Synchronous SnapMirror is not supported on the following systems:

- F87
- F810
- F820

Not all SnapMirror configurations are supported: The following are configurations that are not supported:

- One source system cannot have synchronous SnapMirror relationships to multiple destination systems.
- If using Flexvol volumes, one source cannot be the source of one SnapMirror relationship and the destination of another synchronous SnapMirror relationship. See "Supported SnapMirror configurations" on page 90.
- You cannot have two clustered systems that have bidirectional synchronous SnapMirror relationships between them. See "Supported SnapMirror configurations" on page 90.

Not all data requires synchronous transfers: You decide which data you want synchronously replicated. For example, you might synchronously replicate database data and asynchronously replicate home directories. If the home directories contain important log data, you might synchronously replicate it, but adjust the synchonicity of the transfers to maintain performance levels. Synchronous SnapMirror allows a user-defined lag time of a number of operations or milliseconds before transferring data. See "snapmirror.conf file entry syntax" on page 119 and the na_snapmirror.conf(5) man page for details.

The source filer and destination filer should be adequately configured for the replication traffic: Synchronous SnapMirror is supported only for configurations of which the source system and destination system are the same type of system and have the same disk geometry.

The type of system and disk geometry of the destination system impacts the perceived performance of the source system. Therefore, the destination system should have the bandwidth for the increased traffic and for message logging. Log files are kept on the root volume; therefore, you should ensure that the root volume spans enough disks to handle the increased traffic. The root volume should span four to six disks.

The transport should be optimized for best performance: Synchronous SnapMirror can support traffic over Fibre Channel and IP transports. SnapMirror also allows multipathing, giving you the ability either to

balance the load between two paths or to reserve the second path for failover. For optimizing performance, you can use the best route available between the source and destination system and you can restrict the route to traffic between the two systems. See “Using SnapMirror over multiple paths” on page 126 for details.

Special license key required: In addition to a standard SnapMirror license, the synchronous SnapMirror feature requires a special license key. You must enter this key and receive important product information prior to enabling the synchronous SnapMirror feature.

To enable the synchronous SnapMirror feature, use the `license add` command, and enter one of the following codes.

Platform	License
Data ONTAP storage system	KZZTWOJ
NearStore®	TXKMEAK
Gateway	PLFQNUJ

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.

Using DataFabric® Manager to manage SnapMirror software

You can use the DataFabric Manager graphical user interface to perform the following management tasks in a SnapMirror environment:

- Create and manage asynchronous and synchronous SnapMirror relationships
- Create and manage policies for replication and failover
- Report on relationships and lag times
- Alert on replication state changes
- Schedule replica updates
- Visualize relationships
- Simplify data services recovery after a failure

See the *DataFabric Manager Administration Guide* for more information.

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 storage systems. See "Entering license codes" on page 128.
`options snapmirror.access` or /etc/snapmirror allow file	Specifies the host names of storage systems that are allowed to copy data directly from the source system. See "Specifying destination systems on the source" on page 113.

Command or file	Function
/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 116.

`snapmirror on`
or
`options snapmirror.enable`
	Enables SnapMirror on the source and destination storage systems. See “Enabling SnapMirror” on page 128.

`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 130.

`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 130 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 138.
`snapmirror status`	Displays the status of SnapMirror data transfers. See “Checking SnapMirror data transfer status” on page 144.

Command or file	Function
/etc/log/snapmirror	Displays the SnapMirror data transfer history. See "Checking SnapMirror data transfer logs" on page 153.
`snapmirror abort`	Stops a transfer that is in progress. See "Aborting a SnapMirror transfer" on page 158.
/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 163.

SnapMirror shutdown commands

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

Command	Function

`snapmirror off`
or
`options snapmirror.enable off`
	Turns off SnapMirror functionality for a specified system. See "Turning off SnapMirror updates" on page 166.

`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 168.
`snapmirror release`	Releases for deletion SnapMirror Snapshot copies on former source volumes or qtrees. See "Releasing partners from a SnapMirror relationship" on page 171.

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 copy by allowing ongoing data transfers to finish and temporarily preventing new transfers from beginning. This action ensures a manual Snapshot copy of a stable database. See "Stabilizing (quiescing) destinations before a Snapshot copy" on page 160.
`snapmirror resume`	Resumes normal data transfer to a destination after it has been quiesced. See "Resuming transfers after quiescing a destination" on page 162.
`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 174.

/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 190.

`snapmirror store`
and
`snapmirror use`
	These commands copy a volume to local tape and continue the backup on subsequent tapes if necessary. See "Using SnapMirror to copy a volume to local tape" on page 196.

`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 201.

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 systems, you must purchase a license and enter the SnapMirror license code on each system. For information on enabling the SnapMirror license, see "Entering license codes" on page 128.

- 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 section on organizing data using volumes and qtrees in the *Storage Management Guide* for information about how to create volumes.

- For SnapMirror volume replication, the destination volume must run under a version of Data ONTAP the same as or later than that of the SnapMirror source volume.

  If you configure volume SnapMirror to support replication for the purpose of disaster recovery, both the source and destination systems should run the same version of ONTAP software.

  Caution

  If you upgrade your systems to a later version of Data ONTAP, upgrade the systems of SnapMirror destination volumes before you upgrade the systems 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 system must be in the /etc/hosts file of the destination system 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 *Storage Management 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 382 for information about using the `vol status -b` command to check the capacity of the source volume and destination volume. See the *Storage Management Guide* for information about how to add disks to a volume. See "Initializing a SnapMirror destination" on page 130 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 storage system. 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 "->" (this restriction applies to source qtrees too)
  - Contain space or tab characters
  - 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 storage systems.
- Each storage system model supports a specified number of `snapmirror update` or `vol copy` operations at a time. (See "Maximum number of simultaneous replication operations" on page 86 for the number of copy operations each system can support.)

Cautions

Be advised of the following cautions:

- Do not delete Snapshot copies that SnapMirror creates in the source volume and then copies to the destination. The most recent SnapMirror Snapshot copy is referred to as the newest common Snapshot copy (NCS). Incremental changes to the destination depend on the newest common Snapshot copy. If SnapMirror cannot find the required Snapshot copy 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, stagger your Snapshot update schedules so that SnapMirror activity does not begin, or end at the exact minute a `snap sched` operation attempts to create a Snapshot copy.

  If the SnapMirror feature is scheduled to perform Snapshot management at the same time as a `snap sched` activity, then the Snapshot management operations scheduled using the `snap sched` command might fail with syslog messages, "Skipping creation of hourly snapshot," and "Snapshot already exists."

- 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.

Snapshot schedule conflicts

If the SnapMirror feature is scheduled to perform Snapshot management at the same time as a `snap sched` activity, then the Snapshot management operations scheduled using the `snap sched` command might fail generating syslog messages such as, "Skipping creation of hourly snapshot," and "Snapshot already exists."

To avoid this condition, stagger your Snapshot update schedules so that SnapMirror activity does not begin, or end at the exact minute a `snap sched` operation attempts to create a Snapshot copy.

Ensuring accessible destinations when using 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 system 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 in 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.

Adjusting the TCP window size

A TCP window is the amount of data that a source can send on a connection before it requires acknowledgement from the destination that the data was received. The default window size for SnapMirror operations is 1,994,752 bytes. This is a large window and, in some networks, particularly WANs, can result in the termination of the SnapMirror transfer or very low throughput. To change the TCP window size to fit your network better and avoid these problems, complete the following steps.

Step	Action
1	Calculate a TCP window size that works well in your network using the following fomula: WindowSize = (RoundTrip Delay) (DesiredRate) **Example:** If your average round trip delay is 100 milliseconds and you want a rate of 10 Mbps, you should set the TCP window size to 125,000 bytes as the following equation shows: 0.1 * 10,000,000/8 = 125,000

Step	Action
2	Adjust the TCP window size by entering the following command: **`options snapmirror.window_size rate`** *rate* is the desired TCP window size.

Firewall usage

The destination storage system contacts the SnapMirror source storage system at TCP port 10566 using any of the permissible, available ports assigned by the system. Your firewall must allow requests to this port of the SnapMirror source storage system. It is recommended that you allow a range of TCP ports from 10565 to 10569.

Setting up a basic SnapMirror operation

Completing 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 when using CIFS" on page 103.

To complete a basic SnapMirror setup, complete the following steps.

Step	Action
1	Make sure you have purchased a SnapMirror license for both the source and destination systems.
2	In both the source system and the destination system consoles, use the `license add` command to enable the SnapMirror license on the source and destination systems. **`license add snapmirror_license_code`** For more information see "Entering license codes" on page 128.
3	On the source system console, use the `options snapmirror.access` command to specify the host names of systems that are allowed to copy data directly from the source system. For example: **`options snapmirror.access host=d_systemA`** For more information see "How to specify destination systems on the source system" on page 113.

Step	Action
4	Through the Data ONTAP AdminHost, create or edit the /etc/snapmirror.conf file on the destination system to specify the volumes and qtrees to be copied and the schedule (*minute hour day_of_month day_of_week* or `sync`) on which the destination is updated. For example, the following entry specifies Snapshot mirroring from volume 0 of s_systemA to volume 1 of d_systemA at a maximum of 2,000 kilobits per second 15 minutes past every hour, Monday through Friday: **`s_systemA:vol0 d_systemA:vol1 kbs=2000,restart=always 15 * * 1,2,3,4,5`** If you want to synchronously mirror volume 0 to volume 1, you use a command similar to the following: **`s_systemA:vol0 d_systemA:vol1 - sync`** For more information on schedule entries in the destination system's /etc/snapmirror.conf file, see "Defining source and destination through snapmirror.conf" on page 116 and the na_snapmirror.conf(5) man page.
5	In both the source system and destination system console, use the `snapmirror on` command to enable SnapMirror on the source and destination systems. For more information, see "Enabling SnapMirror" on page 128.

Step	Action
6	Prepare the destination system appropriately, depending on whether you are setting up SnapMirror volume or qtree replication. ◆ If you are setting up SnapMirror volume replication, in the destination system console, use the `vol create` command to create a volume to be the destination for the 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 system 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 volume management chapter of the *Storage Management Guide*; and see the information on restricting volumes in "How to initialize a SnapMirror destination" on page 131.

Step	Action
7	In the destination system 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 systemA) to the destination volume (vol2 on systemB). The destination volume must be configured as restricted and read-only. **`snapmirror initialize -S systemA:vol0 systemB:vol2`** **SnapMirror qtree replication example:** Invoking the following command line creates a destination qtree (qtree4 on vol1on systemB) and transfers a complete copy of the qtree source (qtree4 on vol1 on systemA) to that destination qtree. The volume in which the destination qtree is created must be online and writable. **`snapmirror initialize -S systemA:/vol/vol1/qtree4 systemB:/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 130 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 system. For more information, see "Converting a destination to a writable volume or qtree" on page 168.

Converting asynchronous SnapMirror to synchronous SnapMirror

If you have an asynchronous SnapMirror relationship and you want to convert it to a synchronous SnapMirror relationship, complete the following step.

Step	Action
1	On the administration host, edit the snapmirror.conf file on the destination system to change the schedule to `sync`. **Result:** Data ONTAP uses the new schedule to change the asynchronous SnapMirror to a synchronous SnapMirror.

Configuring for Fibre Channel use

If you are using Fibre Channel connections between the source system and the destination system, the systems must use 2352 (X1024) adapters. After configuring the Fibre Channel adapters, you can perform asynchronous and synchronous SnapMirror replication across the Fibre Channel connection. To configure the systems for Fibre Channel use, complete the following steps.

Step	Action
1	Install the Fibre Channel adapters in the source and destination systems. See the hardware and service guide for your storage system for installation instructions and the *System Configuration Guide* to ensure you install the adapter in the correct slot.
2	Connect the systems to Fibre Channel switches. See the hardware and service guide for more information.

Step	Action

3	On the console of each switch, configure the switch for a default configuration with no zoning and add long-distance Extended Fabric keys. See your switch documentation for configuration information.
	
	Note
	Ensure that the Fibre Channel switches are in fabric mode and support the Simple Name Service (SNS) protocol with support for symbolic names.
	
	Example: The following Brocade switch commands might be used to configure a Brocade switch:
	`switch_con> switchdisable`
	`switch_con> configdefault`
	`switch_con> fastboot`

4	Reboot both systems.

5	Configure the Fibre Channel adapter on each system by entering the following command at the console for each system:
	`ifconfig ql`*`Xa`* *`ipaddr`* `netmask 255.255.255.0 up`
	X is the system slot number.
	a is either port a or port b (labeled on the adapter).
	ipaddr is the IP address of the adapter.
	
	Note
	Ensure that IP addresses for the ports on the same fabric have the same net number.
	
	Example: The following commands configure port a of two Fibre Channel adapters. The net address is 10.10.10.0.
	On system A:
	`ifconfig ql3a 10.10.10.24 netmask 255.255.255.0 up`
	On system B:
	`ifconfig ql3a 10.10.20.24 netmask 255.255.255.0 up`

Step	Action

6	Ensure that the two systems can communcate by using the `ping` command.
	
	Example: The IP address on the destination system is 10.10.20.24.
	
	`source> ping 10.10.20.24`
	`10.10.20.24 is alive`

7	Follow the steps in "Completing a basic SnapMirror setup" on page 105 to complete the SnapMirror setup.

How to troubleshoot reset messages: After configuring Fibre Channel adapters, you might see adapter reset messages similar to the following:

```
Sun Sep  7 20:30:00 PDT [system1: cf.nm.nicReset:warning]:
Interconnect nic 2 is being reset
Sun Sep  7 20:30:03 PDT [system1: nic_mgr:info]: Saving reg dump
for Qlogic VI Fibre Channel Adapter(2) to file
/etc/ispfcvi_regdump.5
```

If you see an adapter reset message like this, check whether you have any of the problems listed in the following table and use the recommended solutions as needed.

Problem	Solution
The cable from the Fibre Channel adapter to the Fibre Channel switch is disconnected.	Connect the cable to the switch.
The Inter Switch Link (ISL) connecting the two Fibre Channel switches is disconnected.	Connect the ISL.
The snapmirror.conf file specifies an IP address or host name that is not configured on the Fibre Channel fabric.	Edit the snapmirror.conf file to use the correct IP address or host name.

Problem	Solution
The host name used in the snapmirror.conf file does not match the IP address in the /etc/hosts file on the storage system.	Ensure that the IP addresses used for the hosts on the Fibre Channel fabric are all in the same subnet.

Specifying destination systems on the source

How to specify destination systems on the source system

There are two methods of specifying destination systems on the source system:

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

  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 system, enter the following command: **`options snapmirror.access`** ***`access_specification`*** The syntax for specifying which systems 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 systemA and to another system named systemB, enter the following at the prompt on systemA:

```
systemA> options snapmirror.access host=systemA,systemB
```

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 system.
2	Add the name of each storage system 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 system.
3	Save edits to the file.

Example: If you want SnapMirror to copy data locally on systemA and to other systems named systemB and systemC, create a /etc/snapmirror.allow file on systemA with the following lines added:

```
systemB
systemC
```

Entries in the /etc/snapmirror.allow file are case-sensitive. You can use the `hostname` command on the destination system 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 system. 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 system.

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 system reported by the `hostname` command. When the option is on, the source system 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 system. In this mode, literal IP addresses (for example, 123.45.67.89) and fully qualified names (for example, systemA.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 system. For example, if the request comes from the IP address of a Gigabit Ethernet interface e10 named systemA-e10, then the /etc/snapmirror.allow file must contain “systemA-e10” or “systemA-e10.acme.com” so the name resolves to the correct IP address.

A local SnapMirror relationship, between two volumes on the same storage system, 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 system, 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 storage system.

How to distribute a snapmirror.conf file

You can create a single /etc/snapmirror.conf file for your site and copy it to all the systems that use SnapMirror. The /etc/snapmirror.conf file can contain entries pertaining to other systems. For example, the /etc/snapmirror.conf file on systemB can contain an entry for copying a volume from systemC to systemD. When systemB reads the /etc/snapmirror.conf file, it ignores the entries for other storage systems. However, each time the file is read, a warning message is displayed on the system console for each line that is ignored.

Limitation on entries for each storage system

There is no limit on the total number of entries in the /etc/snapmirror.conf file; however, there is a limit of 600 entries for each system in the /etc/snapmirror.conf file. There is a 150-entry limit for each system for Data ONTAP 6.2 and a 210-entry limit for each system for Data ONTAP 6.3 through Data ONTAP 6.4.2. Entries beyond the entry limit for each system are ignored and a warning message is displayed on the system console.

If you have a cluster, the limit on the number of entries applies to the storage system pair combination. For example, if your cluster is running Data ONTAP 6.4.2, the 210 entry limit is shared by both systems in the cluster. If one system is using 120 entries, the other systems has 90 entries available for its use.

Note

This limitation is different from the maximum number of of simultaneous replications you can have on a storage system. For that information, see "Maximum number of simultaneous replication operations" on page 86.

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 storage system.
2	If you want to add comments to the /etc/snapmirror.conf file, precede the comment with a pound sign (#). **Example:** `# Replicating from systemA`

Step	Action
3	For each destination volume or qtree that is to be located on this storage system, type an entry specifying the source, destination, characteristics and schedule of the data transfer on one line using the following syntax: ***source_system:{source_volume | /vol/volume_name/qtree_name} dest_system:{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 119. **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_system:/vol/source_volume/-*** ***dest_system:/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_system:/vol/src_vol/[src_qtree]
dest_system:/vol/dest_vol[/dest_qtree] [arguments] [sched]
```

The parameters in the entries are shown in the following table.

Parameter	Description
src_system	The name of the storage system 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 system name, it should find the IP address for the source system on the network over which you want the transfer to occur. **Example:** Assume you created a private network connecting a source system and a destination system and you named the interface on the source system systemA-e0. The interface name systemA-e0 is what you enter in the *source_system* 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_system	The host name of the system to which the data is copied. The name you use must be the exact host name of the destination system. The *dest_system* field is case-sensitive. You can use the `hostname` command on the destination system to determine what you enter for this field. **Example:** If the name of the destination system is systemA, enter systemA in the *dest_system* 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
cksum= {none | crc32c}	Selects the checksum algorithm that is used to protect SnapMirror transmitted data.
outstanding= {*x*ops | *x*ms | *x*s}	Determines the synchronicity level of the synchronous SnapMirror. The ops suffix allows *x* number of outstanding write operations before forcing the client to wait for an acknowledgement. The ms and s suffixes allow *x* number of milliseconds and seconds, respectively.
visibility_ interval={*x*s | *x*m | *x*h}	Determines the amount of time before an automatic Snapshot copy is created on the source volume that is synchronously mirrored. With synchronous SnapMirror, changes to the source volume do not show immediately on the destination volume, even though the changes have been mirrored. The changes are shown only after the source system takes an automatic Snapshot copy of the source volume. This happens every three minutes by default. You can change the interval for automatic Snapshot copies, but performance can degrade if you set smaller intervals because more Snapshot copies are taken more often. The smallest interval you can set is 30 seconds. The s, m, and h suffixes specify seconds, minutes, and hours, respectively.
wsize=*size*	Determines the TCP window size used by a connection. Different from the `snapmirror.window_size` option which determines the TCP window size for allSnapMirror replications. Use the formula shown in “Adjusting the TCP window size” on page 103 to calculate a TCP window size that works well for a particular connection. TCP window size is in bytes.

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`
	
	Typing sync instead of the four space-separated fields specifies synchronous replication. See the na_snapmirror.conf(5) man page for more information.

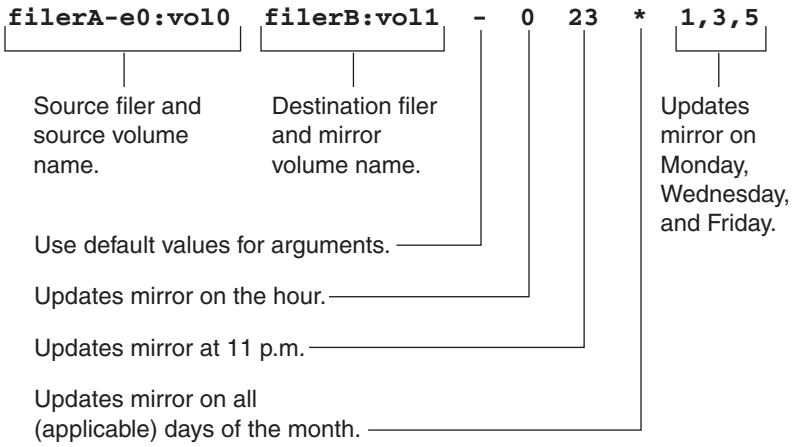
Example of snapmirror.conf schedule entries

Suppose you create a private network between systemA and systemB. In the /etc/hosts file on systemA, you give the host name for the interface as systemA-e0, and you ensure that systemA-e0 is also in the /etc/hosts file on systemB, the destination system. You want to copy vol0 of systemA to vol1 of systemB 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:

```
systemA-e0:vol0 systemB: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.

```
systemA:vol0 systemA: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`.

```
systemA:vol0 systemA: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.

```
systemA:vol0 systemA: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:

```
systemA:vol0 systemA: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:

```
systemA:vol0 systemA: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 copy 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 Snapshot copies 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.

Using SnapMirror over multiple paths

Two modes of multiple paths

You might need more than one physical path for a mirror. SnapMirror supports up to two paths for a particular SnapMirror relationship. The paths can be Ethernet, Fibre Channel, or a combination of Ethernet and Fibre Channel. The two paths can be used in one of two modes:

- Multiplexing mode—SnapMirror uses both paths at the same time, essentially load balancing the transfers. If one path fails, the transfers occur on the remaining path. After the failed path is repaired, the transfers resume using both paths.
- Failover mode—SnapMirror uses the first specified path as the desired path and use the second specified path only after the first path fails.

Setting up SnapMirror using multiple paths

To implement multiple paths between the source and destination storage system, complete the following steps.

Step	Action	

1	Ensure that you have two valid paths using the `ping` command from the source system to each of the IP addresses on the destination system.
	
	Example: The IP addresses on the destination system are 10.10.10.23 and 10.10.10.24.
	
	`source> ping 10.10.10.24`
	`10.10.10.24 is alive`
	`source> ping 10.10.10.23`
	`10.10.10.23 is alive`	

2	**If you are...**	**Then...**
	Setting up SnapMirror for the first time and want to use multiple paths	Follow the steps shown in "Setting up a basic SnapMirror operation" on page 105, then follow Step 3 and Step 4.
	Converting a single path SnapMirror	Follow Step 3 and Step 4.

Step	Action
3	On the administration host, edit the snapmirror.conf file on the destination system to add a connection name line that defines the mode of the connection and what the two connections are. The format of the line follows: *name = mode (src_system1, dest_system1) (src_system2, dest_system2)* *mode* is either multi or failover. See the na_snapmirror.conf(5) man page for details.
4	In the same snapmirror.conf file, edit the schedule entry to reflect the new connection name as the source system.

Note

Multiple paths are supported by SnapMirror running asynchronously and synchronously. The following are examples of implementing multiple paths using synchronous SnapMirror.

Example 1: You want to synchronously mirror volume vol1 on a storage system called NYC to volume vol1 on a storage system called Newark. To satisfy a business requirement that there be two paths between each synchronously mirrored volume, you have four network interface cards, two in each system. You named the two interfaces on the NYC system NYC-pri and NYC-sec and the two on the Newark system Newark-pri and Newark-sec. To implement multiple paths in failover mode, you edit the snapmirror.conf file on Newark to include the following two lines:

```
NYC-Newark = failover (NYC-pri, Newark-pri) (NYC-sec, Newark-sec)
NYC-Newark:vol1 Newark:vol1 - sync
```

Example 2: If NYC-pri and Newark-pri are Fibre Channel adapters and you want to replicate data using both connections, you follow the procedure to configure Fibre Channel adapters for SnapMirror. See “Configuring for Fibre Channel use” on page 109. Then you edit the snapmirror.conf file on Newark to include the following two lines to implement multiple paths in multiplexing mode:

```
NYC-Newark = multi (NYC-pri, Newark-pri) (NYC-sec, Newark-sec)
NYC-Newark:vol1 Newark:vol1 - sync
```

Enabling SnapMirror

SnapMirror enabling procedures

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

Note

In addition to a SnapMirror license, a special license key is required to enable the synchronous SnapMirror feature. See "Synchronous SnapMirror" on page 93 for license codes and specific information.

Entering license codes

Enter the SnapMirror license code on the system that is to be the source and on the system 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 system and destination system 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.allow file to specify allowed destinations...**
	On the source, enter the following command as a single line: **`options snapmirror.access host=[dest_system1,dest_system2...]`** 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 system, 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.

Guidelines when creating a qtree SnapMirror relationship

The following are guidelines for qtrees in a SnapMirror relationship:

- Establish a qtree SnapMirror relationship between volumes that have the same `vol lang` settings.
- Once you establish a qtree SnapMirror relationship, do not change the language assigned to the destination volume.
- Avoid whitespace (space or tab characters) in names of source and destination qtrees.
- Do not rename volumes or qtrees after establishing a qtree SnapMirror relationship.

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 "Using SnapMirror to copy a volume to local tape" on page 196 and "Initializing a SnapMirror destination via local tape" on page 201 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 when using CIFS" on page 103.

Step	Action
1	If the destination is a volume, is online, and has not been initialized before, enter the following command from the destination system: **`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 system, enter the following command: **`snapmirror initialize [`*`options`*`] [`*`dest_system:`*`] {`*`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_system*:]{*source_volume* | *source_qtree_path*} specifies the source system 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 copy (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 copy. *snapshot_name* cannot be minutely.x, hourly.x, nightly.x, or weekly.x, because these names are reserved for scheduled Snapshot copies. This option is valid only for a qtree. ◆ `-s` *snapshot_name* specifies an existing source qtree Snapshot copy to be transferred. This prevents the normal action of the source creating a Snapshot copy to transfer. SnapMirror does not lock or delete this Snapshot copy. This option is valid only for a qtree SnapMirror replication.

Step	Action
	dest_system is the name of the destination system. The destination can reside on the same system as the source or on another system. *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: Using the following command, SnapMirror transfers a complete copy of the source volume (vol0 on systemA) to the destination volume (vol2 on systemB):

```
systemB> snapmirror initialize -S systemA:vol0 systemB:vol2
```

Example 2: Using the following command, SnapMirror transfers a complete copy of the qtree source (qtree4 on vol1 on systemA) to the destination qtree (qtree4bak on vol1on systemB):

```
systemB> snapmirror initialize -S systemA:/vol/vol1/qtree4
systemB:/vol/vol1/qtree4bak
```

Initializing a destination for non-qtree data

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

- Configuration and logging directories on the system (for example, /etc or /logs) that are not normally visible to system clients
- Directories and files on a storage system 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 system, enter the following command: **snapmirror initialize -S *source_system*:/vol/*source_volume*/-*dest_system*:/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: Using the following command, SnapMirror transfers to the destination qtree (non_qtree_data_in_vol3 on vol4 on systemB) a complete copy of all the data in systemA, volume 3, that is not a part of a qtree:

```
systemB> snapmirror initialize -S systemA:/vol/vol3/-
systemB:/vol/vol4/non_qtree_data_in_vol3
```

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:

```
systemB> 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 copy 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 copy 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 Snapshot copies 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 144. If you specify no options or arguments, the `snapmirror status` command shows the status of the volumes in the system, 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.

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

```
Source         Destination     State          Lag       Status
systemA:vol0   systemA:vol0bak  Snapmirrored  00:56:58  Idle
systemA:vol1   systemB:vol6     Source        23:69:26  Transferri
ng
(126 MB done)
```

Checking on the initialization of a qtree

To check that a destination qtree has been created and initialized, complete the following step.

Step	Action

1	Enter the following command:
	`qtree`
	If you specify no options or arguments, the `qtree` command shows the status of all the qtrees in the storage system, as in the following example:
	
	`systemA> qtree`
	`Volume  Tree  Style Oplocks  Status`
	`--------  -----  ----- -------  ------`
	`vol0  unix  enabled  normal`
	`qtree24  unix  enabled  normal`
	`systemB_vol0  unix  disabled  normal`
	`systemB_vol0  qt1  mixed enabled  snapmirrored`
	`systemB_vol0  qt2  unix  disabled  normal`
	`systemB_vol0  qt3  ntfs  enabled  snapmirrored`

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 Snapshot copies 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 system, enter the following command: **`snapmirror update [`*`options`*`] [`*`dest_system`*`:] {`*`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 copy to be transferred, rather than a Snapshot copy 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 Snapshot copies on the source and then copy them.

Step	Action
	◆ `-c` *snapshot_name* creates a Snapshot copy 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 copy. *snapshot_name* cannot be minutely.x, hourly.x, nightly.x, or weekly.x, because these names are reserved for scheduled Snapshot copies. ◆ `-S` [*source_system:*]*source_volume* | *qtree_path* specifies the source system 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_system* specifies the name of the destination system. *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: Using the following command, SnapMirror updates the destination (vol2 on systemB) from the source specified in the /etc/snapmirror.conf file:

```
systemB> snapmirror update systemB:vol2
```

Example 2: Using the following command, SnapMirror updates the qtree destination on systemB:/vol/vol2/usersbak from the source qtree on systemA:/vol/vol1/users:

```
systemB> snapmirror update -S systemA:/vol/vol1/users
systemB:/vol/vol2/usersbak
```

Creating extra backup Snapshot copies for SnapMirror qtrees

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

Example:

```
systemB> snap create vol2 my_snap

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

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

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 Snapshot copies that are no longer necessary for updating data.

Listing SnapMirror Snapshot copies

About SnapMirror listing

You can use the Data ONTAP `snap list` command to list all Snapshot copies, including the SnapMirror-specific Snapshot copies, that are stored on your storage system.

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

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

Naming conventions for Snapshot copies used by SnapMirror

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

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

A SnapMirror volume Snapshot copy name is in the following format:

dest_system(sysid)_name.number

dest_system is the host name of the destination system.

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 copy, starting at 1. Data ONTAP increments this number for each transfer.

A SnapMirror qtree Snapshot copy name is in the following format:

dest_system(sysid)_name-src|dst.number

dest_system is the host name of the destination system.

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 copy. Data ONTAP increments this number for each transfer.

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

Volume example:

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

Qtree example:

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

Caution

Do not delete manually created Snapshot copies 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 Snapshot copies that are created on a source volume and copied to a destination volume. In this example, data is copied from vol1 of systemA (the source) to vol2 of systemB (the destination).

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

If the administrator were to run the `snap list` command on the destination systemB after the systemB(0016782130)_vol2.1 Snapshot copy was transferred from systemA to systemB, a listing similar to the following example would be generated:

```
systemB> snap list vol2
working.....
```

```
%/used    %/total  date          name
--------  -------  ------------  --------
0% ( 0%) 0% ( 0%)  Nov 17 10:50  systemB(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 copy is created on systemA.

The `snap list` command on systemA would generate the following display after the systemB(0016782130)_vol2.2 Snapshot copy was created on systemA:

```
systemA> snap list vol1
working....
%/used   %/total     date          name
-------- ----------  ------------  --------
0% ( 0%) 0% ( 0%)    Nov 17 10:52  systemB(0016782130)_vol2.2
(snapmirror)
0% ( 0%) 0% ( 0%)    Nov 17 10:51  systemB(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 systemB(0016782130)_vol2.2 Snapshot copy is transferred from systemA to systemB, both Snapshot copies exist on systemB. On systemA, however, systemB(0016782130)_vol2.1 is no longer needed and is deleted; only systemB(0016782130)_vol2.2 is retained to be used for the next transfer.

You can see a list of each SnapMirror Snapshot copy 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 storage system.

What checking status shows you

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

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

You see a status report for any SnapMirror source that contains the base Snapshot copy, 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 copy are listed.

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

```
systemA> 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`*`] [[`*`system:`*`] [`*`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 a Snapshot copy" on page 160. *system* is the name of the source storage system. *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 copy was created and transferred successfully, as shown in Example 1.

Example 1: With no options, the information displayed by the `snapmirror status` command might look like the following. See the table "What the categories in the command output mean" on page 148 for more information.

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

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

```
systemB> snapmirror status -l

Snapmirror is on.
Source:                  systemA:vol0
Destination:             systemB:vol2
Status:                  Syncing
Progress:                 60 KB
State:                   Source
Lag:                     00:01:17
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:      1052 KB
Last Transfer Duration:  00:00:02
Last Transfer From:      systemA:vol0

Source:                  systemC:vol0
Destination:             systemB: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:           system2(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:      systemC:vol0

Source:                  systemC:vol4
Destination:             systemB: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:           systemB(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:     systemB:vol4

Source:                 systemC:/vol/vol1/qt5
Destination:            systemB:/vol/vol4/qt5
Status:                 Idle
Progress                -
State:                  Quiesced
Lag:                    0:05:12
Mirror Timestamp:       Sun Jul 16 05:56:12 GMT 2000
Base Snapshot:          systemB(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:     systemC:/vol/vol1/qt5

Source:                 systemC:/vol/vol2/qt1
Destination:            systemB:/vol/vol4/qt2
Status:                 Quiescing
Progress                -
State:                  Snapmirrored
Lag:                    0:02:33
Mirror Timestamp:       Sun Jul 16 05:58:20 GMT 2000
Base Snapshot:          systemB(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:     systemC:/vol/vol2/qt1
```

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

```
systemC> 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 entries on the source category are as follows.

Source entry	Description
s*ystem:vol*	The source system and source volume
system:qtree_path	The source system 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.
system:tape_device	The source tape device; transfer from this tape device is still in progress
base snapshot	The name of the base Snapshot copy 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 entries on the destination category are as follows.

Destination entry	Description
system:vol	The destination system and volume
system:qtree_path	The destination system and qtree path
system: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 192.

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 copy is still present in the volume. If the Snapshot copy 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 Snapshot copies 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 system and the destination is on another system, 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 copy last successfully transferred to the destination.

Lag entry	Description
-	The destination is not initialized.

SnapMirror status entries on the status category are as follows.

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 copy.
Waiting	SnapMirror is waiting for a new tape to be put in the tape device.

Additional snapmirror status entries are as follows.

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 copy successfully transferred from the source to the destination. **Note** A resynchronization may change the base Snapshot copy to a Snapshot copy with an older timestamp.
Base Snapshot copy	The name of the base Snapshot copy 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 copy for that qtree on the destination. **Note** A resynchronization might change the name of the base Snapshot copy.
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 system 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 system for which you want the information: **`options snapmirror.log.enable`** **Result:** SnapMirror reports whether logging is enabled or not.

Example:

```
systemA> 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 system for which you want the log: **`options snapmirror.log.enable on`** **Result:** SnapMirror enables the logging of transfers for the system. This setting is persistent across reboots.

Where the log files are kept

SnapMirror keeps the current log on the root volume of the executing system, 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_system`*`:`*`source_path dest_system`*`:`*`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 storage system 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 system1:vol1 system2:vol1 Request (10.56.17.133)
src Fri Jul 27 20:41:32 GMT system1:vol1 system2:vol1 Abort (Destination not allowed)
src Fri Jul 27 20:45:31 GMT system1:vol0 system1:vol1 Request (10.56.17.132)
src Fri Jul 27 20:45:35 GMT system1:vol0 system1:vol1 Start
src Fri Jul 27 20:51:40 GMT system1:vol0 system1:vol1 End (26200 KB)
src Fri Jul 27 22:41:09 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB Request
(10.56.17.133)
src Fri Jul 27 22:41:12 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB Start
src Fri Jul 27 22:41:13 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB Abort (Non
unicode directory found in source qtree.)
src Fri Jul 27 22:45:53 GMT system1:/vol/vol1/qtb system2:/vol/vol1/qsmb Request
(10.56.17.133)
src Fri Jul 27 22:45:56 GMT system1:/vol/vol1/qtb system2:/vol/vol1/qsmb Start
src Fri Jul 27 22:45:59 GMT system1:/vol/vol1/qtb system2:/vol/vol1/qsmb End (3800 KB)
cmd Fri Jul 27 22:50:29 GMT system1:/vol/vol1/qtb system2:/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 system1:vol0 system1:vol1 Request (Initialization)
dst Fri Jul 27 22:50:20 GMT system1:vol0 system1:vol1 Abort (Destination is not
restricted)
dst Fri Jul 27 22:57:17 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB Request
(Initialize)
dst Fri Jul 27 22:57:24 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB Start
dst Fri Jul 27 22:57:36 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB End (55670 KB)
dst Fri Jul 27 23:10:03 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB Request
(Scheduled)
dst Fri Jul 27 23:10:07 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB Start
dst Fri Jul 27 23:10:18 GMT system1:/vol/vol1/qtA system2:/vol/vol1/qtB End (12900 KB)
cmd Sat Jul 28 00:05:29 GMT - system2:/vol/vol1/qtB Quiesce_start
cmd Sat Jul 28 00:05:29 GMT - system2:/vol/vol1/qtB Quiesce_end
cmd Sat Jul 28 00:05:40 GMT - system2:/vol/vol1/qtB Break_command
cmd Sat Jul 28 00:41:05 GMT system1:/vol/vol1/qtA system2:/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 system 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
systemA:vol0    systemA: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 system, enter the following command: **snapmirror abort [-h] {[*dest_system*:]*dest_volume* | [*dest_system*:]/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_system* is the name of the destination system. *dest_volume* is the destination volume. /vol/*volume_name/qtree_name* is the path name of a destination qtree. If no destination system is specified, the local host's name is used for the system name. You can enter more than one destination volume. You can obtain the destination system 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:
```
systemA> snapmirror abort vol1 systemB:vol2
systemC:/vol3/qtree3

snapmirror abort: Aborting transfer to vol1 systemB:vol2
systemC:/vol3/qtree3
```

Result: SnapMirror aborts the transfer to vol1 on systemA, where the command was entered, and aborts the transfer to vol2 on systemB and the transfer to qtree 3 in vol3 on systemC.

Stabilizing (quiescing) destinations before a Snapshot copy

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 copy of a SnapMirror destination volume or qtree that contains a database, you may need to ensure that its contents are stable during the Snapshot copy. 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 system 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: `systemA> snapmirror quiesce vol1`

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

Result: SnapMirror stops any further data transfers to vol1.

Example 2: `systemA> 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: `systemA> 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.

Note

If you decide to abort a SnapMirror quiesce operation, press Ctrl-C or enter the `snapmirror resume` command at any time.

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 system 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:
	`systemA> 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 storage systems 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 Snapshot copies it no longer needs on the source. See "Converting a destination to a writable volume or qtree" on page 168 and "How to release a source from a SnapMirror relationship" on page 171.

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 copy taken in the source for the data transfer remains, but it can be deleted and replaced by a new Snapshot copy 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:
	`systemA:vol0 systemA:vol1 - 0 23 * 1,3,5`
	`systemA:vol1 systemB:vol6 - 0 23 * 1,3,5`
	Changed update schedule:
	`systemA:vol0 systemA:vol2 - 0 23 * 1,3,5`
	`systemA:vol1 systemB: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:
	`systemA:vol0 systemA:vol1 - 0 23 * 1,3,5`
	`systemA:vol1 systemB:vol6 - 0 23 * 1,3,5`
	`#systemB:vol1 systemC:vol2 - 0 23 *`
	`1,3,5`
	◆ Putting a dash (-) in one of the schedule fields (minute/hour/dayofmonth/dayofweek)
	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.
	Example:
	`systemA:vol0 systemA:vol1 - 0 23 * 1,3,5`
	`systemA:vol1 systemB:vol6 - 0 23 * 1,3,5`
	`systemB:vol1 systemC:vol2 - - 23 *`
	`1,3,5`

Turning off SnapMirror updates

Why you turn off updates

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

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

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

Step	Action
1	Enter the following command on both the source system and destination system 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 copy 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 system alone does not affect SnapMirror on the source system. Other systems can continue to copy data from the source system. **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 storage systems, and you want the same quota restrictions to be applied after you make the destination writable, the destination system must have an /etc/quotas file that includes all the entries from the /etc/quotas file used by the source system.

- If you use SnapMirror replication for data migration, you can copy the /etc/quotas entries from the source system to the /etc/quotas file of the destination system 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 system of all /etc/quotas entries used by the source system at all times. That way, you can apply the quota

entries to the destination volume or qtree if the source system 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 system 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 system. **`snapmirror break volume_name`** ◆ To make a destination qtree writable, enter the following commands on the destination system. **`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 system 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** If 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 136.

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 171.
- Use the `snapmirror resync` command to restore or redefine the SnapMirror relationship. See "Resynchronizing SnapMirror" on page 174.

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 storage systems.

- On the source system, you use the `snapmirror release` command. Releasing a source from a destination volume or qtree allows the source to delete its base Snapshot copy for the SnapMirror relationship. See "How to release a source from a SnapMirror relationship" below.
- On the destination system, 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 172.

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 system, enter the following command: **`snapmirror release {`*`source_volume`*` | `*`qtree_path`*`} [`*`dest_system`*`:]{`*`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_system* is the name of the system 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 system, SnapMirror uses the name of the system on which you entered the command. **Result:** SnapMirror frees all resources on the source system that had been dedicated to the SnapMirror relationship.

Volume example: `systemA> snapmirror release vol0 systemB:vol2`

Qtree example: `systemA> snapmirror release vol/vol1/qtree2 systemB:/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 storage system and `snapmirror release` on the source system. Unless these extra steps are taken, the Snapshot copies associated with the broken relationship remain stored on the destination system, 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 system to break the SnapMirror relationship between the source and destination objects. For details see "Converting to a writable volume or qtree" on page 169. ◆ invoke the `snapmirror release` command on the source system to release the source object from this SnapMirror relationship. For details see "How to release a source from a SnapMirror relationship" on page 171.
2	On the destination system, invoke the `snapmirror status -l` command to determine which Snapshot copy 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 copy basename associated with the SnapMirror relationship that you just broke.
3	On the destination system, delete the Snapshot copy set that you displayed in Step 2. Enter: `snap delete` *`dest_vol snapshot_basename`*

Step	Action
4	Through the Adminhost client, edit the /etc/snapmirror.conf file on the destination system. 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 119.

Resynchronizing SnapMirror

Why you 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 copy (NCS) shared by the two volumes or qtrees, and removes all newer information on the storage system on which the command is run.

Note

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

Resynchronization will cause the loss of all data written to the destination after the base Snapshot copy 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.

Note

To support data integrity in a restrictive SnapLock™ Compliance environment, the `resync` operation saves all data that was written after the common Snapshot copy to a directory on the volume. For details see "Qtree SnapMirror resync for SnapLock Compliance volumes" on page 213.

Resynchronizing a SnapMirror relationship

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

Step	Action
1	From the destination system, enter the following command: **snapmirror resync [*options*] [*dest_system*:]{*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 copy on the source and on the destination that can be used as the newest common Snapshot copy (base Snapshot copy) 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_system*:]{*source_volume* | *qtree_path*}specifies the system 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_system is the name of the destination system.

dest_volume or /vol/qtree_path 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 copy, which will be used as the base for resynchronization, and generates a list of Snapshot copies on the destination volume that
◆ are newer than the base Snapshot copy and will be deleted
◆ are older than the base Snapshot copy and have already been deleted from the source

Note
For qtree resynchronization, only the common Snapshot copy 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 Snapshot copies on the destination volume (if you are resynchronizing volumes)</td><td>Type <code>y</code> at the prompt.

Result: SnapMirror
◆ Deletes the listed Snapshot copies 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 copy that was created after the common Snapshot copy 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: `systemB> snapmirror resync systemB: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: `systemB> snapmirror resync -S systemA:/vol/vol2/qtreeBob systemB:/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 copy on the source and destination exists

If SnapMirror cannot find a common Snapshot copy 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 reinitialize 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.
- restore 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 128.	
3	On the destination system, 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 system, 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 storage systems.

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

- temporarily make systemB:volB the source and systemA:volA the destination to restore mirrored data back to systemA:volA and to update systemA:volA.
- restore systemA:/vol/volA and systemB: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 copy before the source was disabled and all the data written to systemB:vol/volB after it was made writable is preserved. Any data written to systemA:vol/volA between the last SnapMirror Snapshot copy and the time that systemA:vol/volA was disabled is not preserved.

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

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

Migrating data between volumes by using SnapMirror

What SnapMirror does when migrating data

SnapMirror can migrate data between volumes and redirect NFS clients to the new volume without rebooting the storage system or remounting to volume on NFS clients. The migration must be run on two volumes which are currently the source volume and destination volume in a SnapMirror relationship. SnapMirror does the following once you start the migration process:

- Performs a SnapMirror incremental transfer to the destination volume.
- Stops NFS and CIFS services to the source volume.
- Migrates NFS file handles to the destination volume.
- Makes the source volume restricted.
- Makes the destination volume read-write.

What SnapMirror does not do when migrating data

SnapMirror does not do the following:

- Transfer IP addresses, license keys, or quota information. You must remount on the NFS clients unless one of the following is true:
  - You transfer the IP address of the source system to the destination system independently after the migration.
  - The source and destination volumes reside on the same system, in which case, the IP address to access either volume is the same.
- Migrate CIFS clients. You must reestablish CIFS client sessions after migrating data to the destination volume.

Migrating data

To migrate data to another volume, complete the following step.

Step	Action
1	Enter the following command: **`snapmirror migrate [`*`srcsystem`*`:]`*`srcvolume`* **`[`*`dstsystem`*`:]`*`dstvolume`* *srcsystem* is the source system. *srcvolume* is the source volume. *dstsystem* is the destination system. *dstvolume* is the destination volume.

Moving volume SnapMirror or qtree SnapMirror sources

About moving sources

Whether you are moving a volume SnapMirror source or qtree SnapMirror source to new storage systems or newer drives, as long as there is a Snapshot copy in common on the source and destination, the transition goes smoothly. Volume SnapMirror transfers all of the Snapshot copies as part of the SnapMirror process. Qtree SnapMirror has only one Snapshot copy in common, and different destinations will have no Snapshot copies in common; different destinations will not have Snapshot copies in common unless you replicate them.

When to move sources

In a production environment, you should perform the process of moving SnapMirror relationships from one volume or storage system to another only in a maintenance or out-of-service window. Ensure that new data is not added to the original source during the move.

Terms used when moving SnapMirror sources

The following terms are used in the task description for moving volume SnapMirror sources:

- oldsource—The original filer on which the source resides.
- newsource—The filer to which you are moving the source.
- destination—The filer to which the source is replicated.
- oldsourcevol—The original source volume.
- newsourcevol—The new source volume to which you are moving.
- destinationvol—The volume to which the source is replicated.

For detailed information

The following sections discuss moving volume SnapMirror sources and qtree SnapMirror sources:

- "Moving volume SnapMirror sources" on page 185
- "Moving qtree SnapMirror sources" on page 187

Moving volume SnapMirror sources

Moving from one volume SnapMirror source to another

To moving a volume SnapMirror source volume to another source volume, complete the following steps.

Note

Ensure that the major version of Data ONTAP on the destination storage system is greater than or equal to the major version running on the new source filer. See "Prerequisites to running SnapMirror" on page 100 for more information.

Step	Action
1	Copy the original source to the new source using the following command: `newsource> snapmirror initialize -S oldsource:oldsourcevol newsource:newsourcevol` **Result:** This might take some time to finish.
2	It is recommended that you make oldsource read-only before continuing.
3	Create a manual Snapshot copy on the oldsource filer, using the following command: `oldsource> snap create oldsourcevol common_Snapshot`

4	Update newsource and destination based on oldsource using the following commands:
	`newsource> snapmirror update -S oldsource:oldsourcevol newsource:newsourcevol`
	`destination> snapmirror update -S oldsource:oldsourcevol destination:destinationvol`
	Result: The common_Snapshot copy will be on all volumes because all Snapshot copies are mirrored using volume SnapMirror.

Step	Action

5	Quiesce and break the SnapMirror relationship between oldsource and destination, and oldsource and newsource, using the following commands:
	`destination> snapmirror quiesce destinationvol`
	`destination> snapmirror break destinationvol`
	`newsource> snapmirror quiesce newsourcevol`
	`newsource> snapmirror break newsourcevol`

6	Using an editor, update the /etc/snapmirror.conf file on the destination for the new relationship by replacing the oldsource information with newsource information: **Was:** `oldsource:oldsourcevol destination:destinationvol restart=always 0 * * * *` **Edit to:** `newsource:newsourcevol destination:destinationvol restart=always 0 * * * *`
7	Establish the new SnapMirror relationship, using the following command: `destination> snapmirror resync -S newsource:newsourcevol destination:destinationvol` **Result:** The new SnapMirror relationship will automatically pick the newest Snapshot copy in common to mirror. This will be common_Snapshot. **Note** The SnapMirror relationship will discard any Snapshot copies older than common_Snapshot: namely, the ones used for the last SnapMirror update. This is expected and no data will be lost if you ensured that no new data was added to the oringial source volume during the move.
8	Verify that the SnapMirror relationship is resynchronizing, using the following command: `destination> snapmirror status`

Moving qtree SnapMirror sources

The process for moving qtree SnapMirror sources

Unlike volume SnapMirror, in which all of the Snapshot copies from the source are mirrored to the destintation, with qtree SnapMirror you must create a Snapshot copy on the source and force its propagation to the destination and new source. The process to move the qtree SnapMirror source is to create a Snapshot copy on the original source and then replicate the Snapshot copy on the destinations, both the new source and the existing destination. After this is done, the Snapshot copy will be common on all volumes, allowing for the SnapMirror relationship to be broken from the original source and established between the new source and the existing destination.

Moving from one qtree SnapMirror source to another

To move a qtree SnapMirror source to another source, complete the following steps.

Note

Ensure that the major version of Data ONTAP on the destination storage system is greater than or equal to the major version running on the new source filer. See "Prerequisites to running SnapMirror" on page 100 for more information.

Step	Action
1	Copy the original source to the new source, using the following command: `newsource> snapmirror initialize -S oldsource:/vol/oldsourcevol/qtree newsource:/vol/newsourcevol/qtree` **Result:** This might take some time to finish.
2	It is recommended that you make oldsource read-only before continuing.
3	Create a manual Snapshot copy on the oldsource filer, using the following command: `oldsource> snap create oldsourcevol common_Snapshot`

Step	Action
4	Update the mirrors, using the following commands: **newsource> snapmirror update -c common_Snapshot -s common_Snapshot _S oldsource:/vol/oldsourcevol/qtree newsource:/vol/newsourcevol/qtree** **destination> snapmirror update -c common_Snapshot -s common_Snapshot _S oldsource:/vol/oldsourcevol/qtree destination:/vol/destinationvol/qtree** **Result:** The -s option of the `snapmirror update` command synchronizes newsource with oldsource and destination with oldsource based on common_Snapshot. The -c option of the `snapmirror update` command creates the common_Snapshot Snapshot copy on the destination filers.
5	Quiesce and break the SnapMirror relationship between oldsource and destination, and oldsource and newsource, using the following commands: **destination> snapmirror quiesce /vol/destinationvol/qtree** **destination> snapmirror break /vol/destinationvol/qtree** **newsource> snapmirror quiesce /volnewsourcevol/qtree** **newsource> snapmirror break /volnewsourcevol/qtree**
6	Using an editor, update the /etc/snapmirror.conf file on the destination for the new relationship by replacing the oldsource information with newsource information: **Was:** **oldsource:/vol/oldsourcevol/qtree destination:/vol/destinationvol/qtree restart=always 0 * * * *** **Edit to:** **newsource:/vol/newsourcevol/qtree destination:/vol/destinationvol/qtree restart=always 0 * * * ***

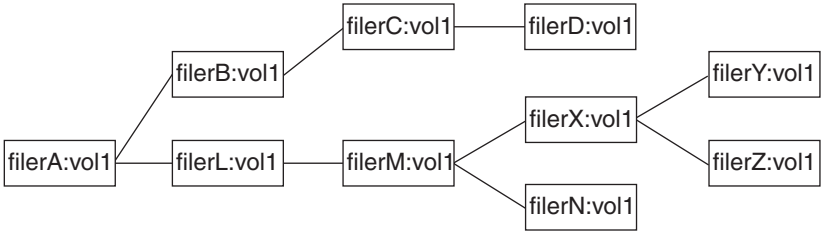
Step	Action
7	Establish the new SnapMirror relationship using the following command: **`destination> snapmirror resync -S newsource:/vol/newsourcevol/qtree destination:/vol/destinationvol/qtree`** **Result:** The new SnapMirror relationship will automatically pick the newest Snapshot copy in common to mirror. This will be common_Snapshot. **Note** The SnapMirror relationship will discard any Snapshot copies older than common_Snapshot: namely, the ones used for the lastest SnapMirror update. This is expected and no data will be lost if you ensured that no new data was added to the oringial source volume during the move.
8	Verify that the SnapMirror relationship is resynchronizing, using the following command: **`destination> snapmirror status`**

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 systemA is copied on nine different storage systems, but only two of the systems copy the data directly from the source. The other seven storage systems 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.

Supported cascading configurations

Because there are different types of SnapMirror available to you, it is not clear what types of SnapMirror are supported as you cascade from one storage system to another. The supported cascading configurations are listed in the following table.

filerA	synchronous volume SnapMirror	filerB	asynchronous volume SnapMirror	filerC
filerA	asynchronous volume SnapMirror	filerB	asynchronous volume SnapMirror	filerC
filerA	asynchronous volume SnapMirror	filerB	asynchronous qtree SnapMirror	filerC
filerA	asynchronous qtree SnapMirror	filerB	asynchronous volume SnapMirror	filerC

This table should be read from left to right. For example, the first line states that filerA has a synchronous volume SnapMirror relationship with filerB and that filerB has an asynchronous volume SnapMirror relationship with filerC.

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 128.

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 systems would look like this:

```
systemA:vol1 systemB:vol1 - 15 * * 1,2,3,4,5
systemA:vol1 systemL:vol1 - 15 * * 1,2,3,4,5
systemB:vol1 systemC:vol1 - 25 * * 1,2,3,4,5
systemC:vol1 systemD:vol1 - 35 * * 1,2,3,4,5
systemL:vol1 systemM:vol1 - 25 * * 1,2,3,4,5
systemM:vol1 systemX:vol1 - 35 * * 1,2,3,4,5
systemM:vol1:systemN:vol1 - 35 * * 1,2,3,4,5
systemX:vol1 systemY:vol1 - 45 * * 1,2,3,4,5
systemX:vol1 systemZ: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. 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 Snapshot copies for cascading destinations

SnapMirror retains the Snapshot copies on the original source volume needed for transfers to destinations further down the line. Snapshot copies that are still needed by a destination are labeled "snapmirror" in the output of the `snap list` command. SnapMirror deletes the Snapshot copies 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 Snapshot copies associated with that destination.

Cascading from a synchronous SnapMirror

You can cascade volume destinations from a synchronous Snapmirror destination, but the cascading series is slightly different from that for asynchronous SnapMirror. For a synchronous SnapMirror, the first replication is the only synchronous SnapMirror replication in the cascade, and the synchronous replication can be to one destination storage system only. Subsequent SnapMirror replications cascading from that destination system must be asynchronous and can be to multiple destination systems.

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 system 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 Snapshot copies 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.

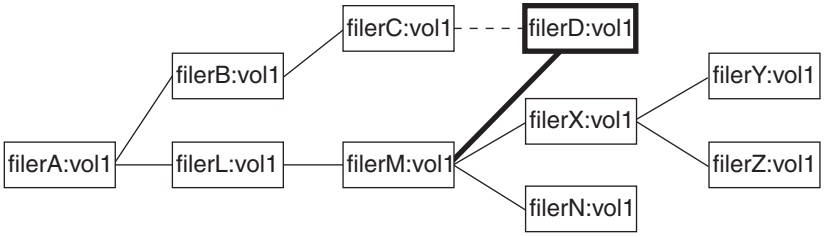
```
systemA> snapmirror destinations vol1

Path        Destination
/vol/vol1   systemB:vol1->systemC:vol1->systemD:vol8
/vol/vol1   systemL:vol1->systemM:vol1->systemX:vol1->systemY:vol1
/vol/vol1   systemL:vol1->systemM:vol1->systemX:vol1->systemZ:vol1
/vol/vol1   systemL:vol1->systemM:vol1->systemN:vol1
```

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

Restructuring a cascade

You might restructure a cascade to balance loading of your systems; to use a storage system or volume for a different purpose; or to perform upgrades, maintenance, or repairs. For example, in the cascade structure that follows, you might want to make systemD:vol1 a destination of systemM:vol1 instead of a destination of systemC:vol1.



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

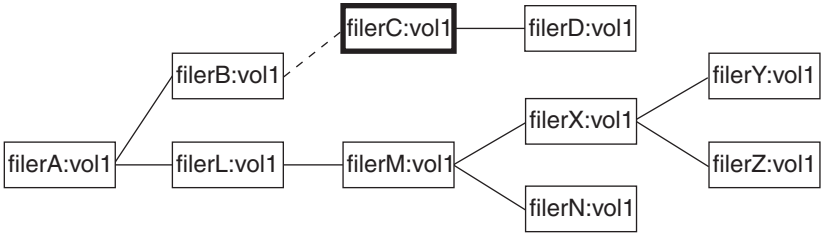
Step	Action
1	On the destination system, change the /etc/snapmirror.conf file to indicate the new source for the destination. **Example:** `systemM:vol1 systemD:vol1 - 35 * * 1,2,3,4,5`

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="2">2</td><td>If...</td><td>Then</td></tr>
<tr><td>The newest Snapshot copy on the destination exists on the source</td><td>Use the following command to update the destination from the new source:
snapmirror update -S source_volume dest_system:dest_volume
Example: <code>snapmirror update -S systemM:vol1 systemD:vol1</code></td></tr>
<tr><td>3</td><td>The newest Snapshot copy on the destination does not exist on the source</td><td>Perform one of the following tasks:
1. Update the new source from the original source using the <code>snapmirror update</code> command. Wait for the destination to update.
2. Make the destination writable using the <code>snapmirror break</code> command. Then resynchronize the destination with the new source using the <code>snapmirror resync</code> command.</td></tr>
<tr><td>4</td><td colspan="2">Release the former source using the following command:
snapmirror release source_volume [[dest_system:]dest_volume]
Example:
<code>systemC> snapmirror release systemC:vol1 systemD:vol1</code></td></tr>
</table>

Example of disconnecting a destination from a cascading series:

For the example shown in the diagram that follows, suppose that from systemB you enter the following command:

```
snapmirror release vol1 systemC:vol1
```



These results follow:

- systemA:vol1 continues to be the source for the destination systemB:vol1.
- systemC:vol1 no longer copies from systemB:vol1. SnapMirror retains Snapshot copies for systemC and below.
- If systemC requests an update from systemB, the destination is reestablished if it is still not writable and the base Snapshot copy still exists on the source.
- systemD:vol1 still copies systemC:vol1.
- All the destinations that depend on systemL:vol1 continue functioning as before.

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

```
systemA> snapmirror destinations -s systemA:vol1
Volume Snapshot                      Destination
vol1   systemB(0015269532)_vol1.37   systemB:vol1
vol1   systemL(0015269532)_vol1.42   systemL:vol1->systemM:vol1-
>systemXvol1->systemY:vol1
vol1   systemL(0015269532)_vol1.42   systemL:vol1->systemM:vol1-
>systemXvol1->systemZ:vol1
vol1   systemL(0015269532)_vol1.42   systemL:vol1->systemM:vol1-
>systemN:vol1
```

Note

If you want to permanently release a destination, 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.

Using SnapMirror to copy a volume to local tape

Why use SnapMirror to local tape

You might want use SnapMirror to copy a volume from a source storage system 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 198.
- 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 203.

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.

Considerations before using as a backup method

SnapMirror replication to copy data to local tape is meant as an initialization process for SnapMirror relationships and has limitations when used as a backup menthod. Consider the following limitations before using SnapMirror to tape as a backup method.

Disk geometry: When SnapMirror replicates data to tape, it optimizes writing of the file system based on the disk geometry of either the source or destination file system. The disk geometry of the source file system is used by default, but you can specify the disk geometry of the destination file system using the `snapmirror store -g` command. See the na_snapmirror(1) man page for more information.

If you retrieve a backup tape into a file system that does not match the disk geometry of the storage system used when writing the data on to tape, the retrieve can be extremely slow.

Note

If FlexVol volumes are in place, disk geometry mismatch when restoring the volume is not an issue.

File system version: Data written to tape is a block by block copy of the file system associated with a particular version of Data ONTAP; therefore, when you retrieve data from tape, the destination of the retrieval must use the same or later version of Data ONTAP used when storing data to tape. If you attempt to retrieve data to a destination that uses an older version of Data ONTAP, the retrieval will fail.

Snapshot copy issues: Because the intended purpose of SnapMirror to tape is to initialize mirrors for SnapMirror relationships, it maintains Snapshot copies that future SnapMirror relationships use to perform updates. When used as a backup method, Snapshot copies are created, but never used; therefore, Snapshot copy and data resources are wasted.

Note

You can manage and delete Snapshot copies using the `snapmirror release` and `snapmirror destinations` commands. See the na_snapmirror(1) man page for more information.

Volume types: The traditional volume file system format is different from the FlexVol volume file system format. Because of this difference, a backed up FlexVol volume cannot be retrieved on a traditional volume. Likewise, a backed up traditional volume cannot be retrieved on a FlexVol volume.

Lack of archive support: Some backups are made to be kept for a long time; as such, they become long term storage of old versions of data. Backups using SnapMirror replication to tape are in a format that is only readable by Data ONTAP and the WAFL® file system, and is not meant for long term storage. Keeping a set of tapes for extended periods of time risks the ability to restore them in the future.

Effects of a bad tape: When retrieving a file system from tape, the format of the SnapMirror replication to tape requires the entire file system to be retrieved before the file system is usable. A bad tape when retrieving a file system means that not all of the file system is retrievable; therefore, the file system is not constructible and all of the data is unavailable.

Lack of backup features: Using SnapMirror replication to copy data to tape does not have features that backup software does. For example, features like individual file restore, backup indexing, and incremental backups are not supported.

SnapMirror source-to-tape-to-destination scenario

In this scenario, you want to establish a SnapMirror relationship between a source system and a destination system 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 system via the low-bandwidth connection.

Prerequisites: This scenario assumes the following configuration:

- A low-bandwidth connection between the source and destination systems
- A local tape drive attached to the source system
- A local tape drive attached to the destination system

Caution

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

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

1. On the source system, use the `snapmirror store` command to copy all volume Snapshot copies, including the base Snapshot copy, 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 199.

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

3. On the destination system, 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 system 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 system over the low-bandwidth connection, or edit the snapmirror.conf file to set up an incremental update schedule from the source to destination system.

   For more information, see "Initializing a SnapMirror destination via local tape" on page 201.

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 copy. For more information see “Releasing a SnapMirror source-to-tape relationship” on page 202.

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 consoles of both the source and destination storage systems, 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 system, load the tape into a local attached tape device.

Step	Action
3	At the source system, 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 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.
6	Follow the procedure described in “Initializing a SnapMirror destination via local tape” on page 201.

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 system. See the *System Administration Storage Management Guide* for information about how to create a volume.
2	Put the volume in the restricted state. See the *System Administration Storage Management Guide* for information about how to restrict a volume.
3	Load the tape (made with `snapmirror store`) into the destination system's local tape device.
4	Start the initialization by entering the following command on the destination system: **`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_system:source_volume [dest_system:]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_system*:*source_volume* specifies the source system and volume for the migration. *source_volume* is the volume you want to copy. *dest_system* is the name of the destination system. *dest_volume* is the destination volume. For more information about the `snapmirror update` command, see "Updating a destination manually" on page 138. **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 117.

Releasing a SnapMirror source-to-tape relationship: After a successful update of the destination system over a connection, you no longer require the source-to-tape SnapMirror relationship and its associated Snapshot copy that you

established with the `snapmirror store` command. To end the source-to-tape backup and delete the Snapshot copy from the source system, complete the following steps.

Step	Action

1	In the source system 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_system:vol1 snapmirror_tape_01_15_03_20:05:32 ...`
	`s_system:vol1 d_system: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 snapmirror_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 use SnapMirror to replicate Snapshot copies 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 copy 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 these steps to set up this arrangement.

1. On the SnapVault secondary/SnapMirror source system, use the `snapmirror store` command to copy the base Snapshot copy 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 204.

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 205.

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.

Step	Action
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 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.
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 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 205.

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 Administration Storage Management Guide* for information about how to restrict a volume.
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_system:*]*source_volume* specifies the source system 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 138.
7	After all tape-to-SnapVault retrieve and update operations are complete, use the `snapmirror break` command to turn the restored system 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 copy and copies that to tape.

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

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

If the base Snapshot copy (most recent common Snapshot copy) 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 copy 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 168 and the section on the `snapmirror resync` command in "Resynchronizing a SnapMirror relationship" on page 176.

If there is no Snapshot copy 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.

```
systemB> vol rename volJobak volStatbak
volJbak renamed to volStatbak
you may need to update /etc/exports
systemB> snapmirror status volJobak
Snapmirror is on.
systemB>snapmirror status volStatbak
Snapmirror is on.
```

```
Source          Destination          State          Lag         Status
-               systemB: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 copy.
- 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 storage system is running at its limit of concurrent transfers, and you attempt to initiate more transfers through manual `snapmirror update` command, the system 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”.

Creating SnapLock destination volumes

What SnapLock volumes are

SnapLock volumes are Write-Once-Read-Many (WORM) volumes that you create for data you want to archive permanently. There are two types of SnapLock volumes:

- SnapLock Compliance volume—for strict regulatory environments, such as SEC 17a-4 compliant environments
- SnapLock Enterprise volume—for environments without regulatory restrictions.

For details about licensing and creating SnapLock volumes, see the *Storage Management Guide*.

Synchronous SnapMirror and SnapLock volumes

SnapLock Compliance volumes do not support synchronous SnapMirror. SnapLock Enterprise volumes do support synchronous SnapMirror.

SnapMirror options

You cannot replicate a non-SnapLock volume to a SnapLock Compliance volume. Nor can you replicate a SnapLock Enterprise volume to a SnapLock Compliance volume.

You can replicate a non-SnapLock volume to a SnapLock Enterprise volume. Note that the SnapMirror replication does not automatically perform WORM commits for files transferred in such a configuration. To perform WORM commits, you have to break the SnapMirror replication and perform the WORM commits on all transferred files, either manually or using a script.

You can also replicate a SnapLock Compliance volume or a SnapLock Enterprise volume to a non-SnapLock volume. There are no restrictions on this replication.

The following table summarizes the different SnapMirror replication options using SnapLock volumes.

	Destination volume type		
Source volume type	**Non-SnapLock**	**SnapLock Enterprise**	**SnapLock Compliance**
Non-SnapLock	Allowed	Allowed	Not allowed
SnapLock Enterprise	Allowed	Allowed	Not allowed
SnapLock Compliance	Allowed	Allowed	Allowed

SnapLock destination volume restrictions

A SnapMirror destination volume created as a SnapLock Compliance volume has the following restrictions:

- You cannot use the `snapmirror resync` command to return a former destination volume and resynchronize its content with the source.

  Note

  This restriction applies to volume SnapMirror relationships only. Resynchronization is supported for qtree SnapMirror relationships.

- You cannot reinitialize a SnapLock Compliance destination volume because data on the volume cannot be changed. If you break the SnapMirror relationship between the source and destination volumes using the `snapmirror break` command, Data ONTAP prevents you from ever reestablishing the SnapMirror relationship with the same SnapLock Compliance destination volume.

  You can initialize a new empty SnapLock Compliance destination volume.

A SnapMirror destination volume created as a SnapLock Enterprise volume is the same as a destination volume created for a non-SnapLock volume. You can perform all the same administrative tasks on a SnapLock Enterprise destination volume that you can perform on a non-SnapLock destination.

Qtree SnapMirror resync for SnapLock Compliance volumes

The SnapMirror `resync` command enables you to re-establish a broken SnapMirror relationship without a lengthy baseline transfer. Typically, this command locates the most recent Snapshot copy, discards any data written to the destination after that Snapshot copy, and begins to resynchronize content using the common Snapshot copies.

To support data integrity in a restrictive SnapLock Compliance environment, the `resync` operation saves all data that was written after the common Snapshot copy to a directory on the volume. The SnapMirror `resync` operation locates the most recent Snapshot copy, and saves any changes that occured prior to that Snapshot copy to an image and log file. These files are then stored in following directory on the SnapLock Compliance volume:

/etc/logs/snapmirror_resync_archive/volname_UUID_qtree

The image and log files are named *dump_image_YYMMDD_HHMMSS,* and *dump_log_YYMMDD_HHMMSS,* respectively.

Note

The retention period for image and log files is equal to the longest retention period of any data file in the data set. This ensures that the image and log files will not be deleted before the retention period has passed.

If the save operation fails for any reason, the `resync` transfer operation will not proceed.

For detailed information about the SnapMirror `resync` command, see "Resynchronizing SnapMirror" on page 174.

Creating a SnapLock Compliance SnapMirror relationship

To create a SnapMirror relationship from a SnapLock Compliance volume to a SnapLock Compliance destination volume, complete the following steps.

Note

This procedure assumes you have read about basic SnapMirror setup and you have experience with creating SnapMirror relationships. See "Setting up a basic SnapMirror operation" on page 105 for a details about first-time SnapMirror setup.

Step	Action
1	On the administration host, create or edit the snapmirror.conf file on the destination system to specify the SnapLock Compliance volume source and destination. **Example:** The following entry specifies asynchronous mirroring from vol1 of s_system and vol2 of d_system using the default replication arguments. `s_system:vol1 d_system:vol2 - * * * *`
2	On the destination system console, create the destination volume using the `vol create` command. **Example:** **`vol create vol2 2`** **Caution** Do not use the `vol create -L` command to create a SnapLock Compliance volume because volume SnapMirror cannot use it as a destination volume.
3	On the destination system console, mark the volume as restricted using the `vol restrict` command.
4	On the destination system console, create an initial complete (baseline) copy of the source on the destination and start the mirroring process using the `snapmirror initialize -L` command. **Result:** After successfully completing the baseline transfer, Data ONTAP converts the destination volume to a SnapLock Compliance destination volume before bringing it online. **Example:** **`snapmirror initialize -S s_system:vol1 -L d_system:vol2`** **Note** After the destination volume is converted to a SnapLock Compliance destination volume, it will always be a SnapLock Compliance volume and cannot be changed.

Creating a SnapLock Enterprise SnapMirror relationship

Creating a SnapMirror relationship from a SnapLock Enterprise source volume to a SnapLock Enterprise destination volume is the same as creating a SnapMirror relationship for non-SnapLock Enterprise volumes. See “Setting up a basic SnapMirror operation” on page 105 for a details about first-time SnapMirror creation.

Protecting SnapVault secondaries using volume SnapMirror

How SnapMirror protects SnapVault secondaries

Volume SnapMirror protects SnapVault secondaries by creating SnapMirror relationships to migrate data from the volumes on the SnapVault secondary system to volumes on a remote (tertiary) Data ONTAP storage system. SnapMirror provides an exact replica of the SnapVault secondary data on the tertiary system.

Advantage of SnapMirror protection

The advantage of protecting SnapVault secondaries using volume SnapMirror is that softlock support enables you to continue SnapVault relationships between the original SnapVault primary system and the tertiary system, without initial baseline transfers.

For example, if your SnapVault secondary system becomes unusable because of a disaster, you can manually redirect your next SnapVault transfers to the tertiary system instead of the unusable secondary system. Your tertiary system becomes the new SnapVault secondary system, and your SnapVault transfers continue, using the latest Snapshot copy common to both the primary and the tertiary systems.

Migrating SnapVault data using volume SnapMirror

To migrate a volume that contains SnapVault destination qtrees from one secondary system to a tertiary system without having to perform a baseline transfer, complete the following steps.

Step	Action
1	Ensure that you have Open Systems SnapVault baselines of the directory you are migrating. **Example:** In this procedure, assume a baseline of the bno:C:\500MB directory was backed up to r200-old:/vol/old_vol/bno_C_500MB.

Step	Action

2	Using SnapMirror, replicate the volume from the present secondary system to a volume on the new secondary system.
	
	Example: To replicate the old_vol volume from the r200-old secondary system to the new_vol volume on the r200-new secondary system, complete the following steps on the new secondary system (r200-new):
	
	1. Create the new_vol volume.
	`r200-new> vol create new_vol 3`
	
	2. Mark the new_vol volume restricted.
	`r200-new> vol restrict new_vol`
	
	3. Transfer the old_vol volume to the new_vol volume.
	`r200-new> snapmirror initialize -S r200-old:old_vol new_vol`

3	Quiesce and break the SnapMirror relationship between the old secondary system and the new secondary system. **Example:** To quiesce and break the SnapMirror relationship between r200-old and r200-new, complete the following steps on r200-new: **1.** `snapmirror quiesce new_vol` **2.** `snapmirror break new_vol`

4	Check the SnapMirror status and SnapVault status on the new secondary. SnapMirror status should be `Broken-off`. SnapVault status should be `Snapvaulted` to the new volume on the new secondary system.
	
	Example: Perform the following steps from r200-new.
	
	1. `snapmirror status`
	`Source Destination State`
	`r200-old:old_vol r200-new:new_vol Broken-off`
	
	2. `snapvault status`
	`Source Destination State`
	`bno:C:\500MB r200-new:/vol/new_vol/bno_C_500MB Snapvaulted`

Step	Action
5	Confirm that SnapVault configuration information is not present on the new secondary system by using the `snapvault status -c` command. **Example:** Perform the following step from r200-new. **`snapvault status -c`** `Snapvault secondary is ON.`

6	Add SnapVault configuration information to the registry on the new secondary system using the `snapvault start` command.
	
	Note
	This does not start a new baseline, it updates the registry.
	
	Example: Perform the following step from r200-new.
	
	`snapvault start -S bno:C:\500MB r200-new:/vol/new_vol/bno_C_500MB`
	`Snapvault configuration for the qtree has been set.`
	`Qtree /vol/new_vol/bno_C_500MB is already a replica.`

7	Confirm that SnapVault configuration information is present on the new secondary system using the `snapvault status -c` command.
	
	Example: Perform the following step from r200-new.
	
	`snapvault status -c`
	`Snapvault secondary is ON.`
	`/vol/new_vol/bno_C_500MB source=bno:C:\500MB`

8	Test the new SnapVault relationship by manually updating r200-new.
	
	If you are using the command line to manage your environment, continue to the next step; otherwise, you are finished.
	
	Example: Perform the following step from r200-new.
	
	`snapvault update r200-new:/vol/new_vol/bno_C_500MB`
	`Transfer started.`
	`Monitor progress with 'snapvault status' or the snapmirror log.`

9	Recreate any schedules used on the old secondary system to the new secondary system and ensure access permissions are in place.

Data Protection Using SnapVault 5

About this chapter

This chapter discusses how the optional SnapVault online 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 220
- "Planning SnapVault" on page 228
- "Setting up a basic SnapVault backup" on page 231
- "Setting up SnapVault backup on Open Systems platforms" on page 239
- "Configuring the SnapVault secondary storage system" on page 241
- "Managing SnapVault backup of Open Systems platforms" on page 244
- "Enabling SnapVault" on page 254
- "Starting a SnapVault backup relationship" on page 258
- "Scheduling SnapVault Snapshot copy updates" on page 266
- "Checking SnapVault transfers" on page 272
- "Displaying SnapVault Snapshot copies" on page 278
- "Changing settings for SnapVault backup relationships" on page 282
- "Manually updating individual secondary storage system qtrees" on page 284
- "Creating a Snapshot copy manually" on page 286
- "Restoring SnapVault data to the primary storage system" on page 288
- "Aborting SnapVault transfers" on page 293
- "Ending SnapVault backups for a qtree" on page 295
- "Unscheduling SnapVault Snapshot copies" on page 296
- "Releasing SnapVault relationships" on page 297
- "Using SnapVault to back up data to SnapLock volumes" on page 298
- "Protecting your SnapVault backups through SnapMirror" on page 340
- "Turning SnapVault off" on page 343
- "VERITAS NetBackup and SnapVault" on page 344
- "VERITAS NetBackup and NearStore" on page 354

SnapVault overview

What is SnapVault

SnapVault is a disk-based storage backup feature of Data ONTAP that enables data stored on multiple storage systems to be backed up to a central, secondary storage system quickly and efficiently as read-only Snapshot copies. In event of data loss or corruption on a storage system, backed up data can be restored from the SnapVault secondary 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 storage system or NearStore 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 server running AIX, Solaris, HP-UX, Linux, Windows NT, Windows 2000, or Windows 2003, whose data can be backed up to a SnapVault secondary storage system
- Open Systems SnapVault agent—a software module 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 primary system or a directory on an Open Systems primary platform and its corresponding secondary system qtree
- SnapVault Snapshot copy—the backup images that SnapVault creates at intervals on its primary and secondary storage systems. SnapVault Snapshot copies 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 Snapshot copies 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 Snapshot copies through the `snapvault snap sched` command. As incremental Snapshot copies 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 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: Data ONTAP, AIX, Solaris, HP-UX, Linux, Windows NT server, Windows 2000 server, or Windows 2003 server systems

What data gets backed up and restored

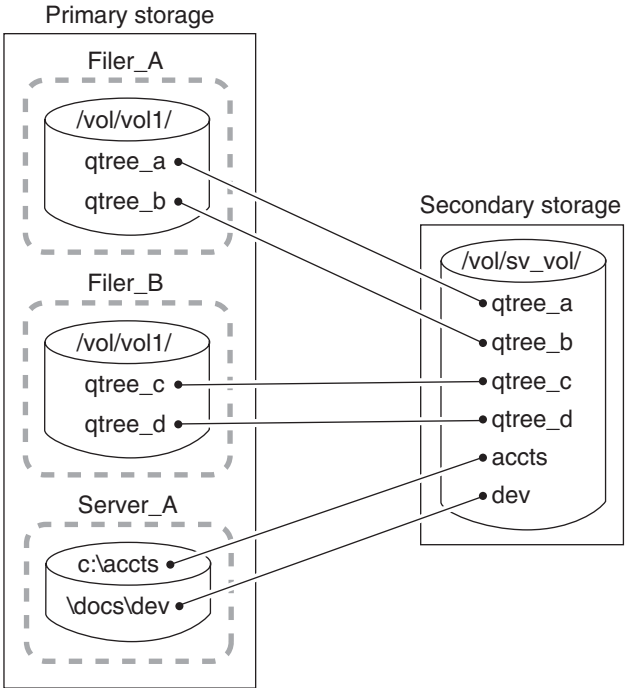
The data structures that are backed up and restored through SnapVault depend on the primary storage system.

- On Data ONTAP storage systems, the qtree is the basic unit of SnapVault backup and restore.

  SnapVault backs up specified qtrees on the primary storage 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 storage systems, 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.

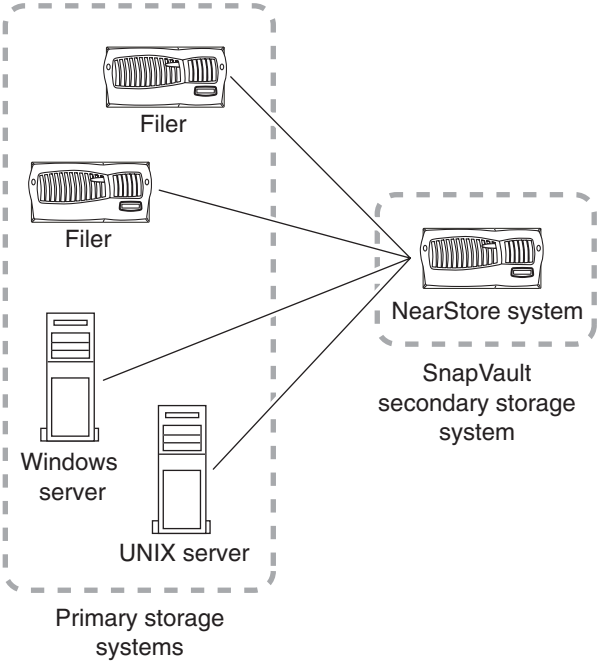
Primary storage systems: Primary storage systems are the Data ONTAP and Open Systems primary storage platforms to be backed up.

- On primary storage systems, SnapVault backs up primary qtree data, non-qtree data, and entire volumes to qtree locations on the SnapVault secondary storage system.
- Supported Open Systems storage platforms include Windows NT servers, Windows 2000 servers, Solaris servers, AIX servers, Linux 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 storage system as Snapshot copies. Any storage system can be configured as a SnapVault secondary storage system; however, the recommended hardware platform is a 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 storage platforms are not subject

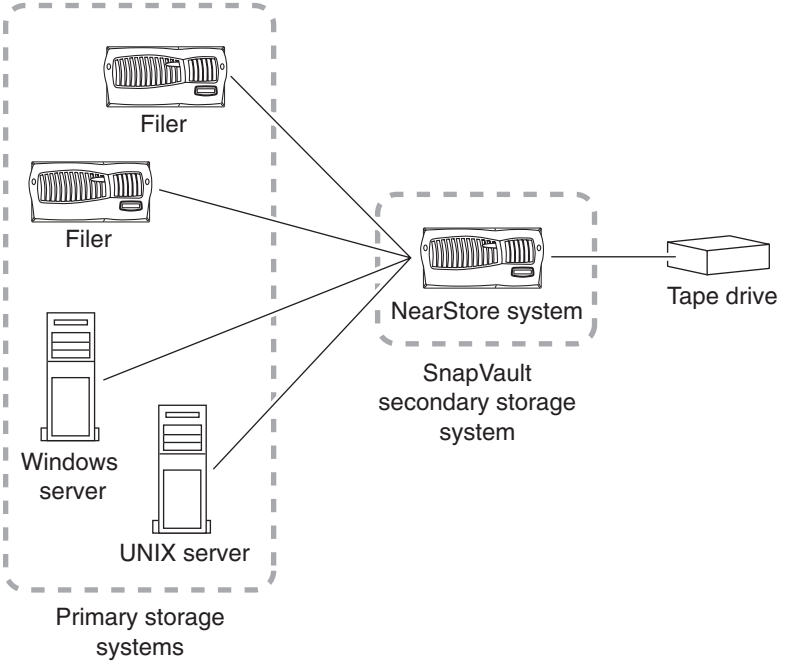
to the performance degradation, system unavailability, and complexity of direct tape backup of multiple systems.

- 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.

Note

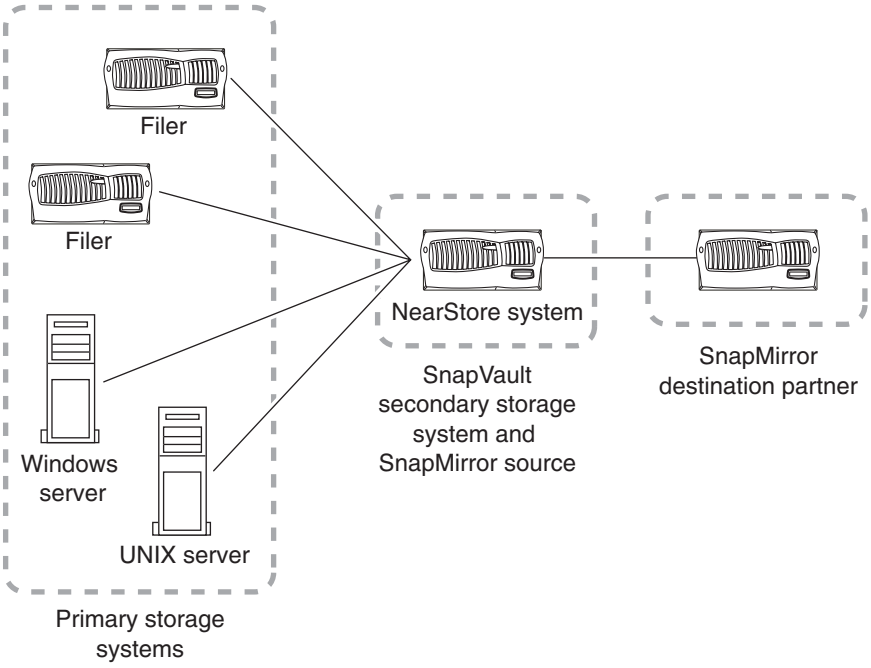
Some Windows NT and UNIX attributes are not preserved using this method; notably, Windows NT sparse files, Windows NT EFS Encrypted Data, and UNIX access control lists (ACLs).



Primary to secondary to SnapMirror variation

Another variation to the basic SnapVault deployment protects Snapshot copies 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 storage systems works

The process of SnapVault backup of storage system 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 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 primary storage system, in response to command line input, creates sets of scheduled SnapVault Snapshot copies (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 Snapshot copies 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.

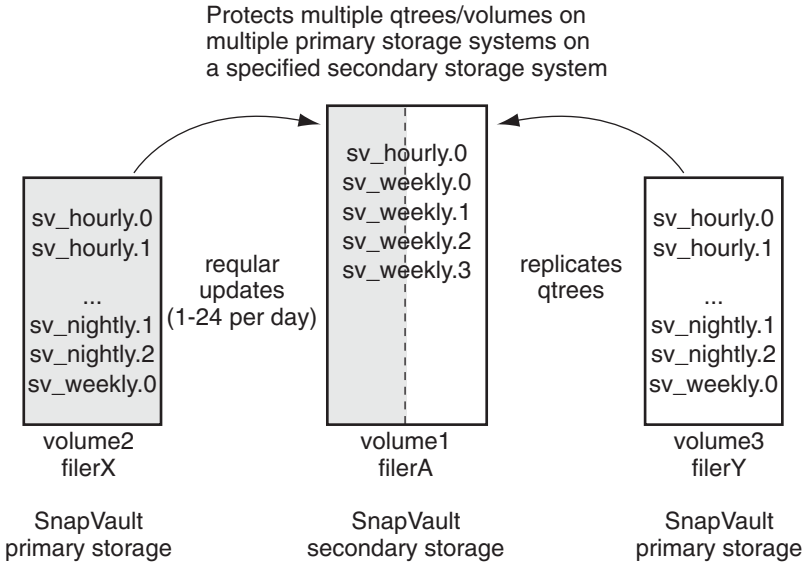
  Then SnapVault creates a volume Snapshot copy of the changes in the secondary qtrees.

  For each transfer and Snapshot set, SnapVault saves the number of secondary storage Snapshot copies that you specify and assigns each Snapshot copy 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 Snapshot copies that capture the changes to the secondary qtrees after each transfer.

  For each set of Snapshot copies, the SnapVault secondary storage system saves the number of secondary storage Snapshot copies you specify and assigns each Snapshot copy 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 Snapshot copies and transfers the data back to the Open Systems storage platform that requests it.

Maximum number of simultaneous SnapVault backups

SnapVault backups have the same limit of simultaneous replications that SnapMirror replications have. See "Maximum number of simultaneous replication operations" on page 86.

Planning SnapVault

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
systemA:/vol/vol1/qtreeAA	sv_secondary:/vol/sv_vol/qtreeAA
systemA:/vol/vol1/qtreeAB	sv_secondary:/vol/sv_vol/qtreeAB
systemB:/vol/vol1/qtreeBB	sv_secondary:/vol/sv_vol/qtreeBB
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 storage-based primaries and SnapVault secondary storage systems, the data to be backed up is captured and preserved in Snapshot copies.

- On the primary storage systems, plan the intervals at which to create SnapVault Snapshot copies of your primary system qtrees.

  Note

  On Open Systems primary storage platforms, Snapshot copy planning and creation does not apply.

- On your SnapVault secondary system, plan the intervals at which you want to update your secondary system qtrees with data transferred from primary storage platforms, and create SnapVault Snapshot copies to retain that information.
- Plan how to limit the combined total of Snapshot copies retained on any one volume of the SnapVault secondary system to 251 or fewer.

Caution

The combined total of Snapshot copies retained on each volume of the SnapVault secondary storage system cannot exceed 251. If the number of Snapshot copies per volume limit is reached and the old Snapshot copies are not deleted, SnapVault will not create new Snapshot copies on that volume.

Before you start SnapVault configuration, use a table like the one below to plan how many Snapshot copies 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 copy every hour or every two hours or every four hours?
- Nightly

  Do you want to create a Snapshot copy every night or just workday nights?
- Weekly

  How many weekly Snapshot copies is it useful to keep?

Snapshot intervals	Primary storage: when created	Primary storage: Snapshot copies retained	Secondary storage: when created	Secondary storage: Snapshot copies 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 Snapshot copies are being kept in the SnapVault secondary storage system volume.

The limit on Snapshot copies per volume is 251, so the 188 Snapshot copies scheduled in this example do not exceed the volume limit.

If you need to retain more than 251 Snapshot copies: If you determine that you need to retain more than 251 SnapVault Snapshot copies on the SnapVault secondary storage system, you can configure additional volumes on the secondary storage system. Each additional volume can support 251 additional Snapshot copies.

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

About configuring SnapVault Backup

Setting up SnapVault backup on the primary systems means preparing the primary storage systems and SnapVault secondary storage systems to fulfill their backup tasks.

Note

You must have separate SnapVault licenses for the primary storage and the secondary storage to use SnapVault.

- On primary storage systems, use console commands to activate the SnapVault primary license, and specify the SnapVault secondary storage host. See "Configuring a primary storage system for SnapVault" on page 232.
- On the SnapVault secondary storage system, use the 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 233.
- On the primary storage systems, schedule times for local SnapVault Snapshot copies to occur. And on the SnapVault secondary storage system, schedule times for those primary Snapshot copies to be backed up to secondary storage. See "Scheduling SnapVault update backups" on page 235.
- 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 copy. See "In event of data loss or corruption" on page 236.

Note

To set up SnapVault backup on Open Systems storage platforms, see "Setting up SnapVault backup on Open Systems platforms" on page 239.

Configuring a primary storage system for SnapVault

On each primary storage system to be backed up to the SnapVault secondary storage system, log in to that system'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. Enter the following command: **`license add sv_primary_license`** For more information, see "Entering license codes" on page 254.
2	Enable the NDMP service on each primary storage system to be backed up. Enter the following command: **`ndmpd on`** For more information see "Setting the ndmpd option" on page 256.
3	Enable SnapVault on each primary storage system to be backed up. Enter the following command: **`options snapvault.enable on`** For more information, see "Setting the enable option" on page 255.
4	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 system must be able to resolve the host name (*snapvault_secondary)* to an IP address in the /etc/hosts file, or else the system needs to be running DNS or NIS. You can also 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 256.

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. Enter the following command: **`license add sv_secondary_license`** For more information see “Entering license codes” on page 254.
2	Enable the NDMP service on each primary storage system to be backed up. Enter the following command: **`ndmpd on`** For more information see “Setting the ndmpd option” on page 256.
3	Enable SnapVault. Enter the following command: **`options snapvault.enable on`** For more information see “Setting the enable option” on page 255.
4	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 256.

Step	Description
5	For each qtree on the 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 system, volume, and qtree, and the secondary volume and qtree. Note that the secondary storage host is optional. Use the `-S` prefix to indicate the source qtree path. For example, **`snapvault start -S system_a:/vol/vol1/tree_a sv_systemb:/vol/sv_vol/tree_a`** **`snapvault start -S system_a:/vol/vol1/tree_b sv_systemb:/vol/sv_vol/tree_b`** **`snapvault start -S system_a:/vol/vol1/tree_c sv_systemb:/vol/sv_vol/tree_c`** **Note** Enter each command on a single line. For more information see “Initiating the first backup of data from primary storage systems” on page 259.

Scheduling SnapVault update backups

On both the 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 primary storage systems:** On each 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 Snapshot copies on each volume containing the backed-up qtrees. For each set of Snapshot copies, specify volume name, Snapshot copy basename (for example: "sv_hourly," or "sv_nightly," and so on), number of SnapVault Snapshot copies to store locally, and the days and hours to execute the Snapshot copies. 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 copy basename, avoid using "hourly," "nightly," or "weekly." Such naming will conflict with the non-SnapVault `snap sched` Snapshot copies. For more information, see "Scheduling Snapshot copies on the SnapVault primary storage system" on page 266.

Step	Description
2	**On the SnapVault secondary storage system:** For each SnapVault volume Snapshot set that you scheduled on your primary storage systems (see Step 1), use the `snapvault snap sched -x` command line to schedule a set of transfers to and subsequent Snapshot copies 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 copy basenames on the primary and secondary systems must match, but Snapshot copy times and number of stored Snapshot copies 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 copy of this volume for archiving. **Note** SnapVault Snapshot copies 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 copy backups to the SnapVault secondary storage system" on page 268.

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 Snapshot copies.

To restore a qtree to its state at the time of one of its SnapVault Snapshot copies, complete the following steps.

Step	Description
1	In the 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 copy that you want to restore from (if it is not the most recent Snapshot copy) ◆ 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 copy. **`snapvault restore -S secondary_host:/vol/sv_vol/tree_b melzhost:/vol/vol1/tree_b`** **Note** Enter the entire command as a single line. For more information, see "Restoring data to the primary storage system" on page 288.

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** Enter the entire command as a single line. For more information see "Restoring data to the primary storage system" on page 288. **Note** To support data integrity in a restrictive SnapLock Compliance environment, the `snapvault start -r` operation saves all data that was written after the common Snapshot copy to a directory on the volume. For details see "SnapVault resynchronization for SnapLock Compliance volumes" on page 298.

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, AIX, Linux, HP-UX systems, and so on) involves installing special agents on those systems and preparing the SnapVault secondary storage system to fulfill its backup tasks.

See the *Open Systems SnapVault Release Notes* for information about supported systems and installing the SnapVault agent on your system.

Backing up and restoring the agent database

The Open Systems SnapVault agent creates a history database of files and directories that the agent is backing up. You can back up this database and restore it if something happens to the original.

Note

You must have root or admin priveledges on the primary storage system to back up or restore the agent database.

To back up the Open Systems SnapVault agent database, complete the following steps.

Step	Action
1	Using the command-line interface of the Open Systems SnapVault primary storage system, navigate to the snapvault/bin directory. ◆ On Windows systems, the default path is c:\Program Files\ snapvault\bin ◆ On UNIX systems, the default path is /usr/snapvault/bin
2	Back up the agent database using the following command: `./svdb -s` **Result:** The primary system prints a file name to the console.
3	Save the file in a safe location.

To restore the Open Systems SnapVault agent database, complete the following steps.

Step	Action
1	Using the command-line interface of the Open Systems SnapVault primary storage system, navigate to the snapvault/bin directory. ◆ On Windows systems, the default path is c:\Program Files\snapvault\bin ◆ On UNIX systems, the default path is /usr/snapvault/bin
2	Restore the agent database by using the following command: **`./svdb -r filename`** *filename* is the file name printed to the console when you backed up the agent database.
3	Stop and restart Open Systems SnapVault services using the svconfigurator utility. See the *Open Systems SnapVault Release Notes* for details.

After installing the agent

On the SnapVault secondary storage system, use the 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 SnapVault secondary storage system" on page 241 and "Managing SnapVault backup of Open Systems platforms" on page 244 for information about these operations.

Configuring the SnapVault secondary storage system

SnapVault secondary storage system requirement

The SnapVault secondary storage system must be running Data ONTAP 6.4 or later to support backup of systems installed with Open Systems SnapVault.

Configuring the SnapVault secondary storage system

To 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 console of the SnapVault secondary system, enter the following command: **`license add sv_secondary_license`** For more information see "Entering license codes" on page 254.
2	License the SnapVault primary storage system. In the console of the SnapVault secondary system, enter the following command: **`license add ossv_primary_license`**
3	Enable SnapVault. In the secondary storage system console, enter the following command: **`options snapvault.enable on`** For more information see "Setting the enable option" on page 255.
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 256.

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** Enter the entire command 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 name of the secondary storage volume containing the secondary qtrees, a Snapshot copy basename (for example: “sv_hourly” or “sv_nightly”), the number of SnapVault Snapshot copies 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 copy 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.

Managing SnapVault backup of Open Systems platforms

Central management of Open Systems SnapVault agents

Open Systems SnapVault agents for all operating systems can be centrally managed using one of the following management tools:

- DataFabric® Manager 2.2 or later.
- NetVault software with a SnapVault Manager plugin module from Bakbone Software, Inc.

  For more information, see the Bakbone web site at http://www/bakbone.com/.

Where to find status and problem reports

You can find primary storage system operational status and problem reports in a log file called snapvault. The snapvault log file is located in the following directory:

install_dir/snapvault/etc

install_dir is the directory on the primary storage system in which you installed the Open Systems SnapVault agent. On Windows systems, the default location for *install_dir* is the c:\Program Files directory. On UNIX systems, the default location for *install_dir* is the /usr directory.

You can find secondary storage system reports in the /etc/log/snapmirror file in the root volume.

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 primary storage systems.

In many cases, the required input for a command is different, reflecting the fact that Open Systems platforms are not organized in qtrees. Instead of primary qtrees, the user specifies directories or volumes to be backed up from the Open Systems platform.

Where directories and volumes are backed up

Each directory or volume you want backed up from a primary storage system is backed up to its own qtree on the SnapVault secondary storage system.

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}`**
`options snapvault.access`	Specifies the 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`**

Command	Input for Open Systems platforms
`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 copy 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**
`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_system*:] *sec_qtree_path***

Command	Input for Open Systems platforms
`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_system:] sec_qtree_path`**
`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`**
`snapvault restore`	Restores a directory, subdirectory, or file from the secondary system to the Open System platform. **Syntax:** At the Open System platform console, enter **`snapvault restore -S [sec_system:]sec_qtree_path prim_qtree_path`**

For detailed information

The following sections discuss different SnapVault backup managment tasks of open systems platforms:

- "Restoring a directory or file" on page 248
- "Restoring an entire primary storage system" on page 251
- "Restoring files to a primary storage system from tape" on page 252
- "Backing up a database" on page 253

Restoring a directory or file

Methods for restoring data

In the event of data loss or corruption on the Open Systems platform, the administrator can use one of three different methods for restoring a directory or file:

- Copy files from the secondary storage system to the primary storage system.
- Use the `snapvault restore` command.
- Use the DataFabric Manager restore wizard.

Copying files

You can copy files from the secondary storage system to the NFS or CIFS primary storage system if you want to restore something other than an entire qtree, that is, a single file, a small group of files, or a few directories.

You might want to share the SnapVault destination on the secondary storage system with all the primary storage systems all the time. In this way, end users can perform self-service restores without bothering a backup administrator.

Note

Some Windows NT and UNIX attributes are not preserved using this method, notably, Windows NT sparse files, Windows NT EFS Encrypted Data, and UNIX access control lists (ACLs).

Copying files to NFS primary storage systems: To restore data by copying files back to an NFS primary storage system, complete the following steps.

Step	Action
1	Mount the backed-up qtree from the SnapVault secondary storage system to the NFS primary storage system.
2	Use the UNIX `cp` command, or an equivalent command, to copy the desired files from the backup to the directory in which you want them.

Copying files to CIFS primary storage systems: To restore data by copying files back to a CIFS primary storage system, complete the following steps.

Step	Action
1	Create a share from the backed-up qtree on the SnapVault secondary storage system to the CIFS primary storage system.
2	Drag and drop the desired files from the backup to the directory in which you want them..

Using the snapvault restore command

You can use the `snapvault restore` command to restore a directory or file on the Open Systems platform to its state at the time of one of its SnapVault Snapshot copies.

To use the `snapvault restore` command, complete the following step.

Step	Description

1	In the command line interface of the Open Systems platform whose data you want to restore, use the `snapvault restore` 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 copy 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
	Example 1 (Windows):
	`snapvault restore -s sv_daily.0 -S myvault:/vol/sv_vol/melzdir/evidence.doc a:\melzdir\ evidence.doc`
	Example 2 (UNIX system):
	`snapvault restore -s sv_daily.0 -S myvault:/vol/sv_vol/melzdir/evidence.doc /usr/melzdir/evidence.doc`
	Note
	Enter the entire command as a single line.

Using the DataFabric Manager restore wizard

The DataFabric Manager restore wizard leads you through the entire restore process. See the *DataFabric Manager Administration Guide* for details.

Restoring an entire primary storage system

Restoring a primary storage system

You can restore an entire NFS or CIFS primary storage system from a SnapVault secondary storage system, but the restore cannot be to a primary storage system that has a blank hard disk. There must be an operating system on the disk.

To restore an entire primary storage system, complete the following steps.

Step	Action
1	Reinstall the operating system on the primary storage system.
2	Reformat the file system as the original file system was formatted.
3	Install the SnapVault agent. See the *Open Systems SnapVault Release Notes* for details.
4	Restore the backed-up directories using the `snapvault restore` command. See "Using the snapvault restore command" on page 249 for details.

Restoring files to a primary storage system from tape

Restoring from tape

To perform a SnapVault restore to an NFS or CIFS primary storage system from tape, complete the following steps.

Note

Some Windows NT and UNIX attributes are not preserved using this method; notably, Windows NT sparse files, Windows NT EFS Encrypted Data, and UNIX access control lists (ACLs).

Step	Action
1	Mount the tape that has the files that need to be restored.
2	Use the `restore` command to restore from the tape to the SnapVault secondary storage system. See the *Tape Backup and Recovery Guide* for details.
3	Copy the files from the SnapVault secondary storage system to the NFS or CIFS primary storage system. See "Restoring a directory or file" on page 248 for details.

Backing up a database

Backing up a database using Open Systems SnapVault

To use Open Systems SnapVault to back up a database, complete the following steps.

Note

The following procedure assumes that all of the files that make up the database reside in a single directory on the primary storage system.

Step	Action
1	On the primary storage system, put the database into hot backup mode.
2	On the SnapVault secondary storage system, back up the data using the `snapvault update` command.
3	On the primary storage system, after the backup completes, take the database out of backup mode.

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 copy creation.
- Turn on the NDMP service on both the secondary storage system and the primary storage system.
- 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 Data ONTAP storage systems only. For SnapVault backup of Open Systems drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 239 and "Managing SnapVault backup of Open Systems platforms" on page 244.

Entering license codes

Enter the SnapVault secondary storage system license code on the secondary storage system, and enter the separate SnapVault primary storage system license code on the primary storage system.

Note

You cannot license a SnapVault secondary, and a SnapVault primary on the same storage system.

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 the *Storage Management 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 ndmpd option

To set the ndmpd option, complete the following step.

Step	Action
1	On both the primary storage system and the secondary storage system, enter the following command: **`ndmpd on`** This option persists across reboots.

Setting the access option

The `snapvault.access` option controls which storage systems can request data transfers. This option persists across reboots.

On the 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 system must be able to resolve the host name entered as *snapvault_secondary* to an IP address in the /etc/hosts file, or else the system needs to be running DNS or NIS. You can also 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:

```
systemB> options snapvault.access host=systemA
```

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 system must be able to resolve the host name entered as *snapvault_primary* to an IP address in the /etc/hosts file, or else the system needs to be running DNS or NIS. You can also use the literal IP address instead of the host name. The syntax for specifying which systems 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:

```
systemA> options snapvault.access host=systemB,systemC
```

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

Specifying data sets on 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 or volumes 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 or volumes and secondary qtrees by

- Specifying the primary system qtrees or volumes 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

Guidelines when creating a SnapVault relationship

The following are guidelines for volumes and qtrees in a SnapVault backup relationship:

- Establish a SnapVault relationship between volumes that have the same `vol lang` settings.
- Once you establish a SnapVault relationship, do not change the language assigned to the destination volume.
- Avoid whitespace (space or tab characters) in names of source and destination qtrees.
- Do not rename volumes or qtrees after establishing a SnapVault relationship.

For detailed information

The following sections discuss the backup relationships you can start:

- "Backing up qtree data" on page 259
- "Backing up non-qtree data" on page 261
- "Backing up volume data" on page 263

Backing up qtree data

Initiating the first backup of data from 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 console for the secondary storage system, enter the following command: **`snapvault start -S prim_system:prim_qtree_path sec_host:sec_qtree_path`** `-S` *prim_system*:*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) to which the data from the primary system qtree is transferred. If no secondary storage system is specified, the local host's name is used. *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_name* must not exist on the secondary system before you run the `snapvault start` command. For information about `snapvault start` command options, see the na_snapvault(1) man page.

Example:

```
systemB> snapvault start -S systemA:/vol/vol2/qtree3
/vol/vol1/qtree3
```

Result: SnapVault creates a Snapshot copy of qtree3 in vol2 of the primary storage system (systemA), copies the data in it to the secondary storage system (systemB) at vol/vol1/qtree3, and configures the qtree for future updates.

Note

The time required for this baseline transfer is limited by the total amount of primary data to be backed up and by the inode count. SnapVault can carry out initial backup at an approximate rate of 7 million inodes per hour (110,000 inodes per minute). In the first phase of a large transfer, SnapVault creates inodes, and it may appear that no activity is taking place.

Backing up non-qtree data

What non-qtree data is

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

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

Initiating backup of primary system non-qtree data

To replicate and protect non-qtree data in a primary storage system, complete the following step.

Step	Action
1	From the secondary storage system, enter the following command: **`snapvault start -S prim_system:/vol/volume_name/- /vol/volume_name/qtree_name`** `-S` *prim_system*:/`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: `systemB> snapvault start -S systemA:/vol/vol1/- /vol/vol4/non_qtree_data_in_vol7`

Result: SnapVault transfers the non-qtree data on primary systemA to the qtree called non_qtree_data_in_vol7 in vol4 on systemB (the secondary storage system). It also configures the qtree for future updates.

Backing up volume data

Data structure of the backed up data

When you back up a source volume using SnapVault, the volume is backed up to a qtree on the secondary storage system; therefore, any qtrees in the source volume become directories in the destination qtree.

Reasons for backing up a volume using SnapVault

You might want to back up a volume to a qtree using SnapVault for one or more of the following reasons:

- You want to back up a volume that contains many qtrees.
- You want the Snapshot management that SnapVault provides.
- You want to consolidate the data from several source volumes on a single destination volume.

Limitations to backing up a volume to a qtree

There are limitations to backing up a volume to a qtree. You should consider the following limitations before performing a volume-to-qtree backup:

- You lose the qtree as a unit of quota management.

  Quota information from the qtrees in the source volume is not saved when they are replicated as directories in the destination qtree.
- You lose qtree security information.

  If the qtrees in the source volume had different qtree security styles, those security styles are lost in the replication to the destination qtree and are replaced by the security style of the volume.
- It is not a simple process to restore data.

  SnapVault cannot restore the data back to a volume. When restoring data, what was a source volume is restored as a qtree.

  See “Restoring the qtree to the original volume structure” on page 264 for a process to restore data.
- Volume-to-qtree backup is not supported for volumes containing Data ONTAP LUNs.

Initiating backup of primary system volume data

To replicate and protect volume data in a primary storage system, complete the following step.

Step	Action
1	From the secondary storage system, enter the following command: **`snapvault start -S prim_system:/vol/volume_name /vol/volume_name/qtree_name`** `-S` *prim_system*:/`vol`/*volume_name* specifies the volume on the primary storage system whose non-qtree data you want to backup. /`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.

Example: `systemB> snapvault start -S systemA:/vol/vol1 /vol/vol4/vol1_copy`

Result: SnapVault transfers the data in vol1 on primary systemA to the qtree called vol1_copy in vol4 on systemB (the secondary storage system). It also configures the qtree for future updates.

Restoring the qtree to the original volume structure

If you were to use the `snapvault restore` command, the source volume you backed up to a qtree is restored as a qtree on the primary storage system.

To restore the backed-up qtree to the original volume structure with multiple qtrees on the primary storage system, complete the following steps.

Step	Action
1	Re-create all of the qtrees in the volume on the primary storage system using the `qtree create` command. **Example:** The following command creates a project_x qtree in the projs volume: **`pri_system> qtree create /vol/projs/project_x`**

Step	Action
2	Restore the data for each qtree using the `ndmpcopy` command. **Example:** The following command restores data from the backed-up project_x directory on the secondary storage system to the re-created project_x qtree on the primary storage system. **pri_system> ndmpcopy -sa *username:password* system2:/vol/vol1/projs/project_x /vol/projs/project_x**
3	Stop qtree updates and remove the qtree on the secondary storage system using the `snapvault stop` command. **Example:** The following command removes the projs qtree from the secondary storage system: **sec_system> snapvault stop /vol/vol1/projs**
4	Reinitialize a baseline copy of each qtree to the secondary storage system using the `snapvault start` command. **Example:** The following command reinitializes the SnapVault backup: **snapvault start -S rdsystem:/vol/projs /vol/vol1/projs**

Scheduling SnapVault Snapshot copy updates

About SnapVault Snapshot copy update schedules

After you have completed the initial baseline backup of qtrees on the primary storage systems to qtrees on the SnapVault secondary storage system (see "Starting a SnapVault backup relationship" on page 258), you must use the `snapvault snap sched` command to schedule

- Regular SnapVault Snapshot copy 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 Snapshot copies of the volume containing the updated secondary qtrees for archiving by SnapVault

Note

The descriptions and procedures in this section pertain to SnapVault backup of Data ONTAP storage systems 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 239 and "Managing SnapVault backup of Open Systems platforms" on page 244.

Snapshot schedule conflicts

If the SnapVault feature is scheduled to perform Snapshot management at the same time as default `snap sched` activity, then the Snapshot management operations scheduled using the `snap sched` command might fail with syslog messages, "Skipping creation of hourly snapshot," and "Snapshot already exists." To avoid this condition, disable the conflicting times using `snap sched`, and use the `snapvault snap sched` command to configure equivalent schedules.

Scheduling Snapshot copies on the SnapVault primary storage system

To schedule a set of Snapshot copies on the SnapVault primary storage system, you use the `snapvault snap sched` command to specify the volume to create Snapshot copies for, the Snapshot copy basename, how many versions of the Snapshot copies to be retained, and the days and hours to execute this set of Snapshot copies.

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 Snapshot copies. *snap_name* is the basename of a Snapshot set, for example, sv_nightly. The name of this Snapshot copy 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` Snapshot copies. *schedule_spec* is made up of *count*[@*day_list*][@*hour_list*]. *count* is the number of Snapshot copies to retain for this Snapshot set. A zero (0) in this field means no new instance of this Snapshot copy will be created. **Caution** The combined total of Snapshot copies retained for this and other Snapshot sets cannot exceed 251 Snapshot copies per volume. If it does, SnapVault will not create new Snapshot copies. @*day_list* is a comma-separated list that specifies the days on which a new Snapshot copy 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 copy will be created automatically. You can create the Snapshot copy manually. The default value is `mon-sun`. @*hour_list* specifies the hours at which a new Snapshot copy 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 Snapshot copies on volume vol1 of the primary storage system.

```
systemB> snapvault snap sched vol1 sv_weekly 1@sat@19
systemB> snapvault snap sched vol1 sv_nightly 2@mon-fri@19
systemB> snapvault snap sched vol1 sv_hourly 11@mon-fri@7-18
```

Result: SnapVault primary storage system creates Snapshot copies on the specified volume, as follows:

- SnapVault creates *sv_weekly.0* every Saturday at 7:00 p.m., and keeps one copy.
- SnapVault creates *sv_nightly.0* every Monday through Friday at 7:00 p.m., and keeps two copies.
- SnapVault creates *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 copy backups to the SnapVault secondary storage system

To schedule the backup of SnapVault Snapshot copies from the 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 copy of the specified volume is created.
- *sec_vol* is the name of the volume on the secondary storage system for which this Snapshot copy is scheduled.
- *snap_name* is the basename of the set of Snapshot copies 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 copy created for this set is

numbered 0, the number of each previous Snapshot copy in this set is increased by 1, and the oldest Snapshot copy in this set is deleted. The *snap_name* must not be “hourly,” “nightly,” or “weekly” to avoid conflict with regular Data ONTAP `snap sched` Snapshot copies.

- *schedule_spec* is made up of *count*[@*day_list*][@*hour_list*].
  - *count* is the number of Snapshot copies to retain for this set. A zero (0) in this field means no new secondary storage system Snapshot copy 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 Snapshot copies retained for this and other Snapshot sets cannot exceed 251 Snapshot copies per volume. If it does, SnapVault will not create new Snapshot copies.

  - @*day_list* is a comma-separated list that specifies the days on which a new Snapshot copy 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 copy will be created automatically. You can create the Snapshot copy manually.
  - @*hour_list* specifies the hours at which a new Snapshot copy 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

Turn off the SnapVault Snapshot schedule on the primary storage system or the secondary storage system at any time with the `snapvault snap unsched` command. (See “Unscheduling SnapVault Snapshot copies” on page 296.)

Example: The following three `snapvault snap sched` command lines schedule three sets of SnapVault transfer updates and Snapshot copies on volume vol1 of the secondary storage system.

```
systemA> snapvault snap sched -x vol1 sv_weekly 5@sat@21
systemA> snapvault snap sched -x vol1 sv_nightly 5@mon-fri@20
systemA> snapvault snap sched -x vol1 sv_hourly 4@mon-fri@8-19
```

Result: SnapVault transfers qtree data from the primary storage system’s Snapshot copy as follows:

- SnapVault transfers *sv_weekly.0* to the secondary storage system every Saturday at 9:00 p.m., makes a new Snapshot copy with the same name containing 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 copy with the same name containing 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 copy with the same name containing all the transferred data, and keeps four copies.

Note

Snapshot copies 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 primary storage systems enough time to create Snapshot copies before the secondary storage system updates from them.

Scheduling Snapshot copies on the secondary system for archiving

You might want to schedule some Snapshot copies on the secondary storage system that do not require a transfer from the primary storage system. For example, you might want to maintain hourly and nightly Snapshot copies on the primary storage system and you might want to keep only weekly Snapshot copies on the secondary storage system. To schedule Snapshot copies on the secondary storage system without having to create a corresponding Snapshot copy on the primary storage system, complete the following step.

Step	Action
1	Enter the following command: **`snapvault snap sched sec_vol snap_name schedule_spec`** **Note** The `snapvault snap sched` command is used because it waits for any active SnapVault transfers to finish before creating the Snapshot copy.

Example: The following command schedules a weekly Snapshot copy at 11 p.m. on Saturdays and keeps the last five Snapshot copies:

```
snapvault snap sched vol2 sv_weekly 5@sat@22
```

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 Snapshot copies sets for all SnapVault volumes.

Example: `systemA> 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 Data ONTAP storage systems 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 239 and "Managing SnapVault backup of Open Systems platforms" on page 244.

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] [[[`*`system_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 copy basename, status, and schedule for each volume. ◆ `-c` displays the configuration parameters of all SnapVault qtrees on the system. This option can be run only from the secondary storage system. *system_name* is the name of the system 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 storage system 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 information in Example 1 below.

  If [*system_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 shown in Example 2.

  See the table "What the status fields mean" on page 275 for more information about the command output shown in Example 1 and Example 2.
- If the `-s` option is given, the output displays Snapshot copy creation status, as shown in Example 3.
- If the `-c` option is given, the output displays current primary storage system configuration parameter settings 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.

```
systemA> snapvault status
Snapvault client is ON.
```

```
Source                    Destination                      State         Lag       Status
systemB:/vol/vol2/qtree3  systemA:/vol/sv_vol/qtree3        Snapvaulted   00:48:24  Idle
```

Example 2: The `snapvault status -l` command displays more detailed information on the most recent SnapVault transfer and Snapshot copy activity.

```
systemA> snapvault status -l
SnapVault client is ON.

Source:                     systemA:/vol/vol2/qtree3
Destination                 systemB:/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:              systemB(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 lists all the Snapshot copies scheduled on the primary or secondary storage system. Information includes volume, Snapshot copy basename, current status, and Snapshot schedule.

systemA> 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 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.

```
systemA> snapvault status -c
/vol/sv_vol/db_qtree1 source=systemB:/vol/db1/s1 kbs=unlimited
tries=20
/vol/sv_vol/db_qtree2 source=systemC:/vol/db2/s2 kbs=unlimited
tries=20
/vol/sv_vol/qtree1 source=systemC:/vol/users/qtree1 kbs=10000
tries=10
/vol/sv_vol/qtree2 source=systemC:/vol/users/qtree2 kbs=10000
tries=10
/vol/sv_vol/qtree3 source=systemD:/vol/users/qtree1 kbs=10000
tries=10
/vol/sv_vol/qtree4 source=systemD:/vol/db/db_1 kbs=7000 tries=10

systemA> snapvault status -c /vol/sv_vol/qtree3
/vol/sv_vol/qtree3 source=systemA:/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 that might be displayed:
Source	*system:qtree_path*—The source is the primary storage system 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	*system:qtree_path*—The destination is the secondary storage system 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 copy 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 that might be displayed:
Status	Aborting—A transfer is being aborted and cleaned up. Idle—No data is being transferred. Pending—The secondary storage system cannot be updated because of a resource issue; the transfer is retried automatically. Quiescing—The specified qtree is waiting for all existing transfers to complete. The destination is being brought into a stable state. Resyncing—The specified qtree is resynchronizing. Transferring—Transfer has been initiated, but has not yet started, or is just finishing.
Progress	Shows the number of KB transferred by the current transfer, or the restart check point if the status is Idle or Pending.
Mirror Timestamp	*hh:mm:ss* indicates the timestamp of the last Snapshot copy successfully transferred from the primary storage system to the secondary storage system. **Note** A resynchronization (`snapvault start -r`) may change the base Snapshot copy to a Snapshot copy with a timestamp older than the original base.
Base Snapshot copy	The name of the base Snapshot copy 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 copy for that qtree on the storage side. A resynchronization (`snapvault start -r`) may change the name of the base Snapshot copy.
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.

Field	Possible values that might be displayed:
Current Transfer Error	Displays an error message if the latest transfer attempt failed.
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 system and volume or qtree (where the content is transferred from).

Displaying SnapVault Snapshot copies

Why to display SnapVault Snapshot copies and qtrees

You can use the `snap list` command to display a list of Snapshot copies 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 Snapshot copies on the secondary storage system (not just SnapVault Snapshot copies)
- The qtrees in the Snapshot copies
- The primary storage system sources of those qtrees
- The timestamp of the primary storage system Snapshot copy that was the source for the data in the secondary storage system Snapshot copy

Using the `snap list -o` command, you can also list the Snapshot copy timestamps, primary qtree origin (if applicable), and Snapshot copy names stored for an individual qtree.

Note

The descriptions and procedures in this section pertain to SnapVault backup of Data ONTAP storage systems 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 239 and "Managing SnapVault backup of Open Systems platforms" on page 244.

How to display SnapVault Snapshot copies on volumes

To see a list of the Snapshot copies and qtrees on your volumes, complete the following step.

Step	Action
1	On the system 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 Snapshot copies. If no volume name is given, the Snapshot copies on all this system'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 copy in the volume. This output is from a volume used as a SnapVault primary storage system.

```
systemA> 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 copy 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 system used as a SnapVault secondary storage system, you see a list of all the SnapVault Snapshot copies retained on volume sv_vol and the details of the qtrees contained in those Snapshot copies.

```
systemB> 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    systemA:/vol/vol2/qtree1
   qtree2      Replica     Jan 22 20:40    systemA:/vol/vol2/qtree2
   qtreeZ      Replica     Jan 22 20:40    systemA:/vol/vol2/qtreeZ
sv_hourly.1  (Jan 22 16:00)
   qtree1      Replica     Jan 22 16:00    systemA:/vol/vol2/qtree1
   qtree2      Replica     Jan 22 16:00    systemA:/vol/vol2/qtree2
   qtreeZ      Replica     Jan 22 16:00    systemA:/vol/vol2/qtreeZ
sv_hourly.2  (Jan 22 12:00)
   qtree1      Replica     Jan 22 12:00    systemA:/vol/vol2/qtree1
   qtree2      Replica     Jan 22 12:00    systemA:/vol/vol2/qtree2
   qtreeZ      Replica     Jan 22 12:00    systemA:/vol/vol2/qtreeZ
.....
```

This output shows which qtrees are replicas of another qtree, and the timestamp of the source Snapshot copy.

Note

Qtrees that are transitioning appear with a dash (-) instead of a timestamp. In general, you should not attempt to restore Snapshot copy versions of qtrees that are transitioning.

If you specify no arguments, the output shows the information for each Snapshot copy in each volume.

How to list Snapshot copies for qtrees

To see a list of Snapshot copies associated with a qtree and, if applicable, the Snapshot copies' primary qtree origins, complete the following step.

Step	Action
1	On the system 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 names of associated SnapVault Snapshot copies that are retained on the system, for example:

```
systemB> snap list -o /vol/sv_vol/qtree3
working...

Qtree /vol/sv_vol/qtree3
date          source                        name
------------  --------                      --------
Jan 31 18:00  systemA:/vol/vol2/qtree3       hourly.0
Jan 31 17:00  systemA:/vol/vol2/qtree3       hourly.1
Jan 31 16:00  systemA:/vol/vol2/qtree3       hourly.2
Jan 31 15:00  systemA:/vol/vol2/qtree3       hourly.3
Jan 30 14:00  systemA:/vol/vol2/qtree3       hourly.4
Jan 30 13:00  systemA:/vol/vol2/qtree3       hourly.5
Jan 30 12:00  systemA:/vol/vol2/qtree3       hourly.6
Jan 30 11:00  systemA:/vol/vol2/qtree3       hourly.7
Jan 31 10:00  systemA:/vol/vol2/qtree3       hourly.8
Jan 31 9:00   systemA:/vol/vol2/qtree3       hourly.9
Jan 31 8:00   systemA:/vol/vol2/qtree3       hourly.10
Jan 30 20:00  systemA:/vol/vol2/qtree3       nightly.0
Jan 29 20:00  systemA:/vol/vol2/qtree3       nightly.1
Jan 26 16:00  systemA:/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. You might need to make these changes if there are hardware or software changes to your storage systems.

Note

The descriptions and procedures in this section pertain to SnapVault backup of Data ONTAP storage systems 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 239 and "Managing SnapVault backup of Open Systems platforms" on page 244.

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 258.

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_system:]prim_qtree_path] [sec_system:]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_system*:]*prim_qtree_path* specifies the primary storage system for the qtree. [*sec_system*:]*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 systemB:/vol2/qtree3 after the name qtree3 on the primary storage system changes to qtreeBob.

```
systemA> snapvault status

Snapvault server is ON.
Source                    Destination                 State           Lag        Status
systemB:/vol/vol2/qtree3   systemA:/vol/sv_vol/qtree3  Snapvaulted     02:48:24   Idle
```

[The qtree3 on systemB is renamed qtreeBob.]

```
systemA> snapvault modify -S systemB:/vol/vol2/qtreeBob /vol/sv_vol/qtree3

systemA> snapvault status
Snapvault server is ON.
Source                      Destination                 State          Lag       Status
systemB:/vol/vol2/qtreeBob  systemA:/vol/sv_vol/qtree3   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 copy on the primary storage system. You might want to update at an unscheduled time to protect the primary storage system data if one of the following conditions exists:

- 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 copy of a quiesced database.

Note

The descriptions and procedures in this section pertain to SnapVault backup of Data ONTAP storage systems only. For SnapVault backup of Opens System drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 239 and "Managing SnapVault backup of Open Systems platforms" on page 244.

How to update the Snapshot copy 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_system`*`:]`*`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 copy 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 copy that is more recent than the current base Snapshot copy. [*sec_system*:]*sec_qtree_path* is the name of the secondary storage system qtree that you want to update.

Example 1: `systemB> snapvault update /vol/vol2/qtree3`

Result: SnapVault updates the qtree on the secondary storage system (systemB) with the data from a new Snapshot copy of the qtree it creates on the primary storage system (systemA). 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 (systemB) with a particular Snapshot copy of qtree3 on the primary storage system (systemA), you would use the `-s` option to specify the Snapshot copy on the primary storage system, as in the following example.

```
systemB> snapvault update -s my_snap systemB:/vol/vol0/qtree3
```

Result: SnapVault updates the qtree on the secondary storage system (systemB) with the data from a Snapshot copy of qtree3 (my_snap) that you created earlier on the primary storage system (systemA).

Note

The `snapvault update` command does not create a new Snapshot copy on the secondary storage system. Use the `snapvault snap create` command if you want to create a new Snapshot copy on the secondary storage system. See "Creating a Snapshot copy manually" on page 286.

Creating a Snapshot copy manually

Why to create a Snapshot copy manually

You might create a manual (unscheduled) Snapshot copy for the following reasons:

- You anticipate planned downtime or you need to recover from downtime (during which a Snapshot copy 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 284, and you want to immediately incorporate that update into the retained Snapshot copies on the secondary storage system.

Note

The descriptions and procedures in this section pertain to SnapVault backup of Data ONTAP storage systems 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 239 and "Managing SnapVault backup of Open Systems platforms" on page 244.

How to create a Snapshot copy 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 copy. To create a manual Snapshot copy 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 copy to be created will reside. *snap_name* is the basename of the Snapshot copy to create. If there is already a Snapshot copy being created in the volume at the time this command is invoked, this command will be carried out after the other Snapshot copy is completed.

Example: `systemB> snapvault snap create vol1 sv_nightly`

Result: SnapVault creates a new Snapshot copy and, based on the specified Snapshot copy basename, numbers it just as if that Snapshot copy had been created by the SnapVault schedule process. SnapVault names the new Snapshot copy sv_nightly.0, renames the older Snapshot copies, and deletes the oldest sv_nightly Snapshot copy.

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 copy. 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 284.

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 start` 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 to 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 Data ONTAP storage systems only. For SnapVault backup of Open Systems drives and directories, see "Setting up SnapVault backup on Open Systems platforms" on page 239 and "Managing SnapVault backup of Open Systems platforms" on page 244.

Restoring data to the primary storage system

To restore data from the SnapVault secondary storage system to the primary storage system, complete the following steps.

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_system:]sec_qtree_path
[prim_system:]prim_qtree_path</tt>
<tt>-S</tt> [sec_system:]sec_qtree_path specifies the secondary storage system and qtree path from which you want to restore the data.
prim_system is the name of the primary storage system that you want to restore to. If specified, this name must match the name of the host system.
prim_qtree_path is the name of the primary storage system qtree that you want to restore to.
For options to the <tt>snapvault restore</tt> command, see the na_snapvault(1) man page.</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_system:]prim_qtree_path [sec_system:]/vol/sec_volume/sec_qtree_path`** `-r` is required to restart backups from a restored primary storage system. See the na_snapvault(1) man page for details about the `snapvault` command syntax. **Note** To support data integrity in a restrictive SnapLock Compliance environment, the `snapvault start -r` operation saves all data that was written after the common Snapshot copy to a directory on the volume. For details see "SnapVault resynchronization for SnapLock Compliance volumes" on page 298.
5	You have finished. See Example 1.
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_system:]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_system*:]*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 (systemA) requesting data to be restored from the secondary storage system (systemB) and then the secondary storage system restarting the SnapVault backup relationship.

```
systemA> snapvault restore -S systemB:/vol/sv_vol/qtree3
vol/vol1/qtree3

systemB> snapvault start -r -S systemA:/vol/vol1/qtree3
systemB:/vol/sv_vol/qtree3
```

Example 2: This example shows the primary storage system (systemA) requesting data to be restored from the secondary storage system (systemB) and then the secondary storage system canceling the SnapVault backup relationship on both storage systems to release the resources used.

```
systemA> snapvault restore -S systemB:/vol/sv_vol/qtree3
/vol/vol1/qtree3

systemB> snapvault release /vol/sv_vol/qtree3
systemA:/vol/vol1/qtree3
```

Deleting the residual Snapshot copy

When you use the `snapvault restore` command to restore a primary qtree, SnapVault places a residual SnapVault Snapshot copy on the volume of the restored primary qtree. This Snapshot copy is not automatically deleted. If you have configured this volume to retain the maximum 251 Snapshot copies allowed by Data ONTAP, you must manually delete this residual Snapshot copy, or else no new Snapshot copies can be created.

To find and delete this residual Snapshot copy, complete the following steps.

Step	Action
1	To display all Snapshot copies (including the residual Snapshot copy) on the volume of the restored qtree, enter the following command: **`snap list`** ***`primaryvolume`*** The residual Snapshot copy will be distinguished by the following syntax: *primaryhost* (*nvram_id*)*_primaryvolume_restoredqtree*-dst.2 **Example:** prim_system (1880911275)_vol1_mytree-dst.2

Step	Action
2	To delete the residual Snapshot copy, enter the following command: `snap delete` *`primaryvolume extrasnapshotname`* **Example:** **`snap delete vol1 prim_system (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.

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 primary storage system or at the secondary storage system.

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. To abort a primary to secondary storage transfer, complete the following step.

Step	Action
1	At the console of either the primary or secondary storage system, enter the following command: **`snapvault abort [`*`sec_system`*`:]/vol/`*`volx`*`/`*`sec_qtree`*** *sec_system* 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. **Note** If you use the `-h` option (hard abort) with the `snapvault abort` command, you cannot restart the transfer.

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. To abort a secondary to primary storage transfer, complete the following step.

Step	Action
1	At the console of either the primary or secondary storage system, enter the following command: **`snapvault abort [`*`prim_system`*`:]/vol/`*`volx`*`/`*`prim_qtree`*** *prim_system* is the name of the primary storage system. *prim_qtree* is the name of the primary qtree to which the data is being restored. **Note** If you use the `-h` option (hard abort) with the `snapvault abort` command, you cannot restart the transfer.

Aborting SnapVault Snapshot copy creation

To abort the ongoing creation of a SnapVault Snapshot copy on the secondary storage system, complete the following step.

Step	Action
1	At the console of the secondary storage system, enter the following command: **`snapvault abort -s `*`volx sv_snapname`*** The `-s` option specifies that the command aborts the attempt to create a Snapshot copy with basename of *sv_snapname* on volume *volx*.

You can obtain the secondary storage system volume and Snapshot copy 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.

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_system`*`:]`*`sec_qtree_path`* [*sec_system*:]*sec_qtree_path* is the qtree that you no longer want to back up.

Example: `systemB> snapvault stop systemB:/vol/sv_vol/qtree3`

Result: SnapVault stops updating the qtree on the secondary storage system and deletes the qtree. Existing Snapshot copies on the secondary storage system are unaffected, but as new Snapshot copies 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 Snapshot copies on the primary storage system. For more information, see "Releasing SnapVault relationships" on page 297.

Unscheduling SnapVault Snapshot copies

Why you unschedule Snapshot copies

You might want to unschedule a set of SnapVault Snapshot copies if the data in the qtrees you are backing up has been migrated to another location or is no longer useful.

How to unschedule Snapshot copies

To turn off the SnapVault schedule for a set of Snapshot copies 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 Snapshot copies to stop creating and without asking for confirmation. *volume* is the name of the volume to stop creating Snapshot copies on. *snap_name* is the basename of the Snapshot set to stop creating. If no *snap_name* is provided, SnapVault turns off the SnapVault Snapshot schedule for all Snapshot sets in the volume and does not update any more SnapVault Snapshot copies in the volume. If no volume is provided, SnapVault unschedules all SnapVault Snapshot sets and does not update any more SnapVault Snapshot copies in the storage system. If there is already a Snapshot copy being created in the volume, the command fails.

Example: `systemB> snapvault snap unsched vol1 sv_nightly`

Result: SnapVault asks for confirmation. If you confirm the action, SnapVault unschedules all SnapVault Snapshot copies with the basename sv_nightly on vol1 of systemB.

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

How to release SnapVault relationships

To release Snapshot copies and free the space they used, complete the following step.

Step	Action
1	Enter one of the following commands. ◆ On a primary storage system console, enter: **snapvault release *prim_qtree_path* *sec_system*:*sec_qtree_path*** ◆ On the secondary storage system console, enter: **snapvault release *sec_qtree_path* *prim_system*:*prim_qtree_path***

Example:

```
systemB> snapvault release /vol/sv_vol/qtree3
systemA:/vol/vol1/qtree3
```

Using SnapVault to back up data to SnapLock volumes

SnapVault backup to SnapLock volumes

You can use SnapVault to back up data, whether it is Write Once Read Many (WORM) data or not, to SnapLock volumes. To keep data on the SnapLock volume compliant, retention periods and log files on a separate LockVault™ log volume are used. For added data protection, you might create a copy of the SnapVault backup using SnapMirror.

The following process of backing up to SnapLock volumes using SnapVault is similar to the process of backing up to non-SnapLock volumes:

- You back up the primary system data using SnapVault. See "Setting up a SnapVault backup" on page 303.
- If you want a copy of the SnapVault backup, you copy the backup using SnapMirror. See "Protecting your Compliant SnapVault backups with SnapMirror" on page 333.

Note

Even though the process of using SnapVault is similar for SnapLock volumes and non-SnapLock volumes, there are some issues of file naming conventions and WORM management of which you should be aware. These are discussed in the following sections.

SnapVault resynchronization for SnapLock Compliance volumes

The `snapvault start -r` command enables you to re-establish a broken SnapVault relationship without a lengthy baseline transfer. Typically, this command locates the most recent Snapshot copy, discards any data written to the destination after that Snapshot copy, and begins to resynchronize content using the common Snapshot copies.

To support data integrity in a restrictive SnapLock Compliance environment, the `snapvault start -r` operation saves all data that was written after the common Snapshot copy to a directory on the volume. The `snapvault start -r` operation locates the most recent Snapshot copy, and saves any changes that occurred prior to that Snapshot copy to an image and log file. These files are then stored in following directory on the SnapLock Compliance volume:

/etc/logs/snapmirror_resync_archive/volname_UUID_qtree

The image and log files are named *dump_image_YYMMDD_HHMMSS,* and *dump_log_YYMMDD_HHMMSS,* respectively.

Note

The retention period for image and log files is equal to the longest retention period of any data file in the data set. This ensures that the image and log files will not be deleted before the retention period has passed.

If the save operation fails for any reason, the `snapvault start -r` transfer operation will not proceed.

For detailed information

The following sections discuss information that helps you to use SnapVault to back up data to SnapLock volumes:

- "Planning capacity" on page 300
- "Setting up a SnapVault backup" on page 303
- "Managing WORM Snapshot copies from SnapVault" on page 309
- "Managing SnapVault log files" on page 326
- "Protecting your Compliant SnapVault backups with SnapMirror" on page 333

Planning capacity

Aspects of capacity planning

The following are aspects of capacity planning:

- SnapVault secondary storage system volume size
- LockVault log volume size
- Number of qtrees backed up to each volume
- SnapVault schedule

Guidelines for the secondary storage system volume size

The following are conservative guidelines you can use to estimate the size of SnapVault secondary storage system volumes for growth over one year, depending on how often you back up data.

- Three percent (3%) growth every day
- Five percent (5%) growth every week
- Ten percent (10%) growth every month
- Seven percent (7%) monthly growth rate, compounded to 100 percent growth every year

Daily backup example: Assuming a starting baseline transfer of 1 GB and 100 percent growth of new data for the year (1 GB). The changed size due to daily growth if changes are made every weeknight is as follows:

Size due to daily growth = (number of days x .03) - (new data)

= (250 x .03) - 1 GB = 7.5 GB - 1 GB = 6.5 GB of change

Therefore, for 1 GB of source data, you need 1 GB baseline + 1 GB new data + 6.5 GB of changed data = 8.5 GB secondary system volume. In other words, you must be able to scale the secondary system volume to about 8 times the current data set size within one year.

Weekly backup example: Using the same assumptions as the daily backup example, the changed size due to weekly growth if changes are made every week is as follows:

Size due to weekly growth = (number of weeks x .05) - (new data)

= (52 x .05) - 1 GB = 2.6 GB - 1 GB = 1.6 GB of change

Using weekly backups, for 1 GB of source data, you need 1 GB baseline + 1 GB new data + 1.6 GB of changed data = 3.6 GB, or a scaling factor of a little over 3.5 times the current data set size with one year.

Guidelines for LockVault log volume size

Consider the following guidelines to estimate the log volume size:

- Baseline transfer = number of inodes per volume x 256 bytes

  where 256 bytes is an estimated size of the log entry.
- Incremental transfers = number of inodes x .03 x 250 Snapshot copies x 256 bytes, where .03 represents the rate of change.

Use these guidelines to estimate the size of the log data associated with each SnapLock volume and create a log volume equal to the total of the combined estimated sizes.

Note

If the estimate is too low, you can add disks to the volume; alternatively, you can allocate a second log volume if the first becomes full and configure the system to use the new log volume.

Guidelines for number of qtrees backed up to a volume

Use the following guidelines when backing up qtrees to a volume using SnapVault:

- Because of performance constraints, a maximum of 16 primary qtrees should be backed up to a single secondary volume.
- To gain optimum performance, the number of qtrees backed up to a volume should be approximately six.
- If you are backing up volumes to qtrees using SnapVault, a maximum of 16 volumes should be backed up to a volume.

Note

You can mix backed up volumes and qtrees on the same secondary volume, if desired.

Guidelines for scheduling SnapVault transfers

The only guideline for scheduling SnapVault transfers is to not overload the primary storage system or the network. You must consider the overall load and time to complete all transfers.

Setting up a SnapVault backup

About setting up SnapVault Backup

Setting up SnapVault backup means preparing the primary storage systems and SnapVault secondary storage systems to perform backup tasks. These tasks are almost identical to those for non-SnapLock volumes. All SnapVault setup tasks are presented here for completeness.

Note

You must have separate SnapVault licenses for the primary storage and the secondary storage to use SnapVault backup.

- On primary storage systems, use console commands to activate the SnapVault primary license, and specify the SnapVault secondary storage host. See "Configuring a primary storage system for SnapVault" on page 303.
- On the SnapVault secondary storage system, use the console commands to license and enable SnapVault, initialize the compliance clock, configure the LockVault log volume, specify the primary storage systems to back up, and start the initial Snapshot copy backup. See "Configuring the SnapVault secondary storage system" on page 304.
- On the primary storage systems, schedule times for local SnapVault Snapshot copies to occur. And on the SnapVault secondary storage system, schedule times for those primary Snapshot copies to be backed up to secondary storage. See "Scheduling SnapVault update backups" on page 306.

Configuring a primary storage system for SnapVault

On each primary storage system to be backed up to the SnapVault secondary storage system, log in to that system'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 console, enter the following command: **`license add sv_primary_license.`**

Step	Description
2	Enable SnapVault on each primary storage system to be backed up. In the console enter the following command: **`options snapvault.enable on`**
3	Use the `options snapvault.access` command to specify the name of the SnapVault secondary storage system to back up to. Enter the following command: **`options snapvault.access host=`*`snapvault_secondary`*** The system must be able to resolve the host name (*snapvault_secondary)* to an IP address in the /etc/hosts file, or else the system needs to be running DNS or NIS. You can also 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.

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 console of the SnapVault secondary system, enter the following command: **`license add `*`sv_secondary_license`***
2	Enable SnapVault. In the console, enter the following command: **`options snapvault.enable on`**
3	Initialize the compliance clock if you have not already done so. In the console, enter the following command: **`date -c initialize`** **Result:** The system prompts you to confirm the current local time and that you want to initialize the compliance clock.
4	Create the LockVault log volume, a SnapLock volume that contains Files Transferred log files. See “Creating SnapLock volumes” on page 366 for details.

Step	Description
5	Enable the LockVault log volume. Enter the following command: **`options snapvault.lockvault_log_volume volume_name`** *volume_name* is the lockvault log volume. See "Configuring the LockVault log volume" on page 326 for more information. **Note** You must use the name of a previously created SnapLock volume for this command to succeed.
6	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 ...`**
7	For each qtree on the 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 system, volume, and qtree; and the secondary storage host, SnapLock volume, and qtree. Use the `-S` prefix to indicate the source qtree path. For example, to start a baseline copy of qtrees tree_a, tree_b, and tree_c to a SnapLock Compliance volume called sv_vol, use the following commands: **`snapvault start -S system_a:/vol/vol1/tree_a sv_systemb:/vol/sv_vol/tree_a`** **`snapvault start -S system_a:/vol/vol1/tree_b sv_systemb:/vol/sv_vol/tree_b`** **`snapvault start -S system_a:/vol/vol1/tree_c sv_systemb:/vol/sv_vol/tree_c`** **Note** Enter each command on a single line.

Scheduling SnapVault update backups

On both the 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 primary storage systems:** On each 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 Snapshot copies on each volume containing the backed-up qtrees. For each set of Snapshot copies, specify volume name, Snapshot copy basename (for example: "sv_hourly," or "sv_nightly," and so on), number of SnapVault Snapshot copies to store locally, and the days and hours to execute the Snapshot copies. 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 copy basename, avoid using "hourly," "nightly," or "weekly." Such naming will conflict with the non-SnapVault `snap sched` Snapshot copies.

Step	Description
2	**On the SnapVault secondary storage system:** For each SnapVault volume Snapshot set that you scheduled on your primary storage systems (see Step 1), enter the following `snapvault snap sched -x` command to schedule a set of transfers to and subsequent Snapshot copies of the SnapVault secondary storage system: **`snapvault snap sched -x sec_vol snap_name count[@day_list] [@hour_list]`** **Caution** Before scheduling SnapVault Snapshot copies, ensure that the SnapLock default retention period is set correctly or that you explicitly set the retention period when you schedule SnapVault Snapshot copies (see "Managing WORM Snapshot copies from SnapVault" on page 309.) Snapshot copy basenames on the primary and secondary systems must match, but Snapshot copy times and number of stored Snapshot copies 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 copy of this volume for archiving. *count* is the number of Snapshot copies you want to retain for this set. This value is ignored for SnapLock volumes. *@day_list* is a comma-separated list that specifies the days on which a new Snapshot copy is created for this set. *@hour_list* specifies the hours at which a new Snapshot copy is created for this set. **Example:** The following commands schedule transfers and Snapshot copies every Saturday at 8 p.m., every Monday through Friday at 8 p.m., and every hour, Monday through Friday from 7 a.m. to 7 p.m., respectively: **`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`**

Step	Description
3	**On the SnapVault primary and secondary storage system:** If the SnapVault feature is scheduled to perform Snapshot management at the same time as default `snap sched` activity, then the Snapshot management operations scheduled using the `snap sched` command might fail with syslog messages, "Skipping creation of hourly snapshot," and "Snapshot already exists." To avoid this condition, disable the conflicting times using `snap sched`, and use the `snapvault snap sched` command to configure equivalent schedules. To turn off the regular Snapshot schedule, enter the command: **`snap sched` *`volume`* `0 0 0`**

Managing WORM Snapshot copies from SnapVault

Snapshot retention for SnapLock volumes

Normally, SnapVault uses the count field of the SnapVault schedule on the secondary storage system to determine the number of Snapshot copies that are retained. Once the count has been reached, the oldest retained Snapshot copies are deleted when new Snapshot copies are added. For SnapLock volumes, older WORM Snapshot copies cannot be deleted until their retention period has expired. WORM Snapshot copies are not deleted automatically. It is your responsibility to delete WORM Snapshot copies once they expire.

Snapshot names for SnapLock volumes

Snapshot copies created by SnapVault on SnapLock Compliance volumes use a different naming scheme from that used by Snapshot copies for regular volumes. Snapshot names for SnapLock volumes use the time and date of the Snapshot copy creation as a suffix. The time and date are generated by the compliance clock. The following is the format of the Snapshot copy name:

snapname.yyyymmdd_hhmmss_zzz

snapname is the Snapshot copy name specified in the schedule

yyyymmdd is the year, month, and day

hhmmss is the hour, minute, and second

zzz is the current time zone setting

Example:

```
%/used        %/total   date          name
----------  ----------  ------------  --------
  2% ( 2%)    0% ( 0%)  Feb 04 02:20  svhourly.20040104_120502_GMT
  9% ( 3%)    1% ( 0%)  Feb 04 02:15  svhourly.20040104_180601_GMT
```

Retention period for SnapVault created WORM Snapshot copies

You can configure the schedules on the SnapVault secondary storage system to create WORM Snapshot copies with a retention period that you specify. For WORM Snapshot copy creation to occur, the volume must exist on the secondary storage system and it must be a SnapLock volume.

Note

If you change the retention period in the schedule, WORM Snapshot copies created under the previous schedule retain their retention periods and WORM Snapshot copies created under the changed schedule use the new retention period.

SnapVault default setting for the retention period

When you configure a SnapVault Snapshot schedule for a SnapLock volume on the secondary storage system, the Snapshot copies created for that volume are WORM Snapshot copies. By default, SnapVault uses the retention period set by the snaplock_default_period vol option as the retention period for the WORM Snapshot copies. You should ensure that the retention periods you configured when creating a SnapLock Compliance volume are correct.

Caution

The default retention period for SnapVault Compliance volumes is 30 years. Be sure to reset the default retention period unless you want all of the created SnapVault WORM Snapshot copies to have a 30-year retention period.

Specifying WORM Snapshot retention periods

You can specify retention periods for different WORM volume Snapshot copies. To specify the retention period for WORM volume Snapshot copies, complete the following step.

Step	Action
1	Enter the following command: **`snapvault snap sched -x -o retention_period=`*`period sec_vol snapname count@day_list@hour_list`* *period* is the retention period in days (d), months (m), or years (y). *sec_vol* is the name of the volume where the Snapshot copy resides. *snapname* is the name of the Snapshot copy. *count* is the number of Snapshot copies you want to retain for this set although this value is ignored. *@day_list* is a comma-separated list that specifies the days on which a new Snapshot copy is created for this set. *@hour_list* specifies the hours at which a new Snapshot copy is created for this set.

Example: WORM Snapshot copies created on the secondary storage system in the sv_proj SnapLock volume have retention periods of 360 days from the time of their creation. Snapshot copies are created at noon and 8 p.m. every day.

```
snapvault snap sched -x -o retention_period=360d sv_proj sv_hourly
1@12,20
```

Extending the retention period of WORM Snapshot copies

To extend the retention period of a WORM Snapshot copy, complete the following step.

Step	Action
1	Enter the command: **snapvault snap retain *volume snapshot period*** *volume* is the name of the WORM volume. *snapshot* is the name of the Snapshot copy. *period* is the retention period in *count* followed by days (d), months (m), or years (y). See the na_snapvault(1) man page for details.

Example: The following command extends the hourly Snapshot copy on the wormvol volume to 2 years:

```
snapvault snap retain wormvol sv_hourly.20050513_195442_GMT 2y

***WARNING: YOU ARE REQUESTING SNAPLOCK RETENTION OF A SNAPSHOT***

This operation will enforce the retention of the snapshot by
SnapLock for the specified retention period. You will NOT be able
to delete the retained snapshot until this retention period has
been satisfied. The relevant information for confirmation of this
operation is as follows:

        Volume: wormvol
      Snapshot: sv_hourly.20050513_195442_GMT
  Retain until: Fri Feb 17 00:00:27 GMT 2006

Are you REALLY sure you want to retain this snapshot? Y
```

Listing WORM Snapshot copies

WORM Snapshot copies are identified by the word snaplock at the end of each Snapshot copy entry. To list all Snapshot copies, including WORM Snapshot copies, complete the following step.

Step	Action

1	Enter the command:
	snap list *sec_vol*
	sec_vol is the name of the WORM volume.
	Example: The following lists Snapshot copies on the wormvol volume:
	`system> snap list wormvol`
	`Volume wormvol`
	`working...`
	
	`%/used %/total date name`
	`-------- ---------- ------------ --------`
	`0% ( 0%) 0% ( 0%) May 13 19:56 sv_hourly.20050513_195442_GMT (snaplock)`
	`3% ( 2%) 1% ( 0%) May 13 19:51 sv_hourly.20050513_195006_GMT (snaplock)`
	`5% ( 2%) 1% ( 0%) May 13 19:12 sv_hourly.20050513_191043_GMT (snaplock)`

To list Snapshot copies with their retention dates, complete the following step.

Step	Action

1	Enter the command:
	`snap list -l volume`
	volume is the name of the WORM volume.
	Example: The following lists Snapshot copies and retention dates:
	`system> snap list -l wormvol`
	`Volume wormvol`
	`working...`
	
	`snapshot date retention date name`
	`------------------------- ------------------------- --------`
	`May 13 19:56:50 2005 +0000 May 13 19:59:42 2005 +0000 sv_hourly.20050513_195442_GMT`
	`May 13 19:51:07 2005 +0000 May 13 19:55:08 2005 +0000 sv_hourly.20050513_195006_GMT`
	`May 13 19:12:06 2005 +0000 May 13 19:15:43 2005 +0000 sv_hourly.20050513_191043_GMT`

Snapshot copies with a dash (-) in the retention date column are not WORM Snapshot copies; therefore, they do not have retention periods.

Deleting expired WORM Snapshot copies

WORM Snapshot copies that have been retained beyond their retention periods can be deleted. To delete an expired WORM Snapshot copy, complete the following step.

Step	Action
1	Enter the command: **`snap delete volume snapshot`** *volume* is the name of the WORM volume. *snapshot* is the name of the Snapshot copy.

Trying to delete unexpired WORM Snapshot copies will fail.

Example:

```
system> snap delete wormvol slminutely.20040104_122040_GMT
Illegal operation on snapshot locked by SnapLock.
```

Retention of more than 255 SnapVault Snapshot copies

You might need to retain more than the 255 Snapshot copies, which is the limit imposed on volumes by Data ONTAP. Actually, the practical limit for WORM Snapshot copies is about 250 because of a few non-WORM base Snapshot copies and temporary Snapshot copies that are used by SnapVault, and Snapshot copies that are needed for managing SnapMirror relationships if you have the SnapVault secondary volumes protected by SnapMirror.

If you need to retain more than 250 Snapshot copies, you can do so by creating a new volume. As subsequent volumes reach the limit, you create newer volumes. In this manner, you can use multiple volumes to retain a larger number of Snapshot copies.

Two approaches to create a new volume: Two approaches to creating a new volume to retain more Snapshot copies are as follows:

- Create a clone of the volume and continue running SnapVault updates to the new volume.
- Copy one Snapshot copy in the volume to a new volume and continue running SnapVault updates to the new volume.

Note

You cannot retain more than 255 SnapVault Snapshot copies without a new baseline using Open Systems SnapVault because Open Systems SnaVault does not support the `snapvault start -r` command, a command that is needed to restart SnapVault relationships on the new volume.

Advantages of cloning: The following are advantages of the cloning approach:

- Less disk space is used—Initially, the new clone volume will have almost no used disk space. Only changes are recorded on the new volume.
- Speed—Volume cloning is almost instantaneous. Copying data from a Snapshot copy takes time.
- Breaking the relationship is easy—If you wish to break the relationship between the original volume and the clone, you can do so using the `vol clone split` command.

  Note

  At the time of the split, additional disk space will likely be used.

Advantages of copying: The following are advantages of the copy approach:

- Each volume is completely independent—You do not have to keep the old volumes online. If the old volumes are damaged or destroyed, your more recent volumes can still be used.

- SnapVault relationships can be migrated to another machine with volume copying.

Tasks for creating a new volume for Snapshot retention

The following process lists tasks for creating a new volume for Snapshot retention.

1. "Ensure everything on the old volume is in an appropriate state" on page 317.
2. "Unconfigure the SnapVault schedules for the old volume" on page 318.
3. "Create a volume clone or copy the appropriate Snapshot copy to a new volume" on page 319.
4. "Check or set the retention periods on the new volume" on page 321.
5. "Check volume options on the new volume" on page 322.
6. "Restart all SnapVault relationships in the new volume" on page 323.
7. "Reconfigure the SnapVault schedules in the new volume" on page 324.
8. "Run a few snapshot targets in the new volume to ensure everything was migrated successfully" on page 324.
9. "Stop all SnapVault relationships in the old volume" on page 325.

Details of each task are listed in the following sections.

Note

Free Snapshot copies might be required to complete some operations. If you get an error message stating that the system has reached the Snapshot copy limit per volume, you will need to find unnecessary Snapshot copies and delete them.

Ensure everything on the old volume is in an appropriate state: To ensure everything on the old volume is in an appropriate state, complete the following steps.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="4">1</td><td colspan="2">Check that all relationships for the volume that you will clone or copy are <code>Idle</code> and in the <code>SnapVaulted</code> state by entering the following command:
<code>snapvault status</code></td></tr>
<tr><td>If...</td><td>Then...</td></tr>
<tr><td>They are in this state</td><td>Go to Step 3.</td></tr>
<tr><td>They are not in this state, but transfers are proceeding normally</td><td>Wait for the transfers to complete, then repeat this step.</td></tr>
<tr><td></td><td>Transfers are not proceeding or completed</td><td>Go to Step 2</td></tr>
<tr><td>2</td><td colspan="2">Occasionally, the <code>snapvault status</code> command might show a relationship for which there is no permanent configuration information. To check and correct the configuration information, complete the following steps.
1. Enter the following command:
<code>snapvault status -c</code>
2. Compare the output from the <code>snapvault status -c</code> command to the output from the <code>snapvault status</code> command. You should see a one-to-one correspondence. If any relationships are missing, make the output from the two commands consistent.
3. Create the configuration entry for the missing relationship using the <code>snapvault start</code> command or, if the relationship is not needed, use the <code>snapvault stop</code> command.
Note
Using the <code>snapvault stop</code> command destroys the relationship. You should only use this command if the relationship was present by mistake.</td></tr>
</table>

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>3</td><td>Ensure that all qtrees with SnapVault relationships are at the same base Snapshot copy by entering the following command:
<tt>snapvault status -l</tt>
Check the Base Snapshot copy listed for each qtree in the destination volume, to ensure they refer to the same Snapshot copy.
<table><tr><th>If...</th><th>Then...</th></tr><tr><td>They refer to the same Snapshot copy</td><td>You are finished.</td></tr><tr><td>They do not refer to the same Snapshot copy</td><td>Go to Step 4</td></tr></table></td></tr>
<tr><td>4</td><td>Enter the following command to create a new base Snapshot copy for all qtrees in the volume:
<tt>snapvault snap create -w sec_vol ""</tt>
sec_vol is the current secondary volume.
“ “ specifies that no snapvault snap sched Snapshot copy is created, such as <tt>sv_hourly.20050513_195442_GMT</tt>.
Note
All qtrees should be in the <tt>SnapVaulted</tt> state after the transfer initiated by the <tt>snapvault snap create</tt> command completes.</td></tr>
<tr><td>5</td><td>Go back to Step 1 and ensure everything is in an appropriate state.</td></tr>
</table>

Unconfigure the SnapVault schedules for the old volume: Before you unconfigure the schedules, make note of the schedule configuration, including retention periods. Doing so enables you to set up the schedule the same way on the new secondary volume.

To unconfigure the SnapVault schedules, complete the following steps.

Step	Action
1	View the existing schedules by entering the following command: **`snapvault snap sched sec_vol`**

Step	Action
2	Unconfigure the schedules by entering the following command: **`snapvault snap unsched sec_vol`** **Note** If transfers to the secondary volume are currently occurring, the `snapvault snap unsched` command fails. Wait for them to finish, and then repeat the command.
3	Ensure that everything is in an appropriate state by repeating the steps in the section "Ensure everything on the old volume is in an appropriate state" on page 317.

Create a volume clone or copy the appropriate Snapshot copy to a new volume: To create a volume clone, complete the following steps.

Step	Action
1	Ensure that you licensed the volume cloning feature (`flex_clone` license) by entering the following command: **`license`**

Step	Action

2	Create the clone by entering the following command:
	`vol clone create altvol -b sec_vol base_snapshot`
	altvol is the new secondary volume.
	sec_vol is the old secondary volume.
	base_snapshot is the Snapshot copy that you found in Step 3 of "Ensure everything on the old volume is in an appropriate state" on page 317.
	Note
	Clone volumes are in the same aggregate as the parent volume by definition.
	When you create the volume clone, you might see several messages that appear to be errors, but they are not. Examples of possible messages are:
	`Reverting volume altvol to a previous snapshot.`
	`Breaking snapmirrored qtree 1 in volume altvol: base snapshot no longer exists. Use snapmirror resync or initialize to re-establish the snapmirror.`
	`Breaking snapmirrored qtree 2 in volume altvol: WAFL_check broke all snapmirrors in volume wormclone1. Use snapmirror resync or initialize to re-establish the snapmirror.`

When creating a copy of a volume, the copy must be the same type as the old volume (flexible or traditional) and it must be the same compliance type as the old volume (SnapLock Compliance or SnapLock Enterprise).

Note

If the volume you are copying is a flexible volume, the copy does not have to be in the same aggregate as the old volume, nor does it have to be on the same storage system as the old volume.

To create a copy of the old volume, complete the following steps.

Step	Action
1	Create the new volume by entering the following command: **`vol create altvol`** *altvol* is the new volume. **Note** Include the appropriate volume type and sizing information.
2	Put the new volume in a restricted state by entering the following command: **`vol restrict altvol`**
3	Copy the base Snapshot copy from the old volume to the new volume by entering the following command: **`vol copy start -s base_snapshot secsystem:secvol altvol`** **Note** The command assumes that the volume copy is initiated from the system which contains *altvol*.

4	Put the new volume online by entering the following command:
	`vol online altvol`
	When you create the volume copy, you might see several messages that appear to be errors, but they are not. Examples of possible messages are:
	`Breaking snapmirrored qtree 1 in volume altvol: base snapshot no longer exists. Use snapmirror resync or initialize to re-establish the snapmirror.`
	`Breaking snapmirrored qtree 2 in volume altvol: WAFL_check broke all snapmirrors in volume wormclone1. Use snapmirror resync or initialize to re-establish the snapmirror.`

Check or set the retention periods on the new volume: Clone volumes and volume copies automatically inherit the retention periods from their parent volumes. For this reason, this step is usually a check. You can change the retention periods for the volume if you want.

To check the retention periods on the new volume, complete the following steps.

Step	Action
1	To see the retention periods on the old volume, enter the following command: **vol options *sec_vol***
2	To see the retention periods on the new volume, enter same command for the new volume: **vol options *altvol***
3	Compare the output from the two volumes to ensure they are the same.
4	If you want to change the retention periods, enter one or more of the following commands: **vol options *altvol* snaplock_default_period *period*** **vol options *altvol* snaplock_maximum_period *period*** **vol options *altvol* snaplock_miniumum_period *period*** *period* is the retention period. See the na_vol(1) man page for retention period values.

Check volume options on the new volume: As was the case with retention periods, clone volumes and volume copies automatically inherit all options from their parent volumes. Before starting any SnapVault activity on the new volume, ensure that this is the case.

To check volume options on the new volume against volume options on the old volume, complete the following steps.

Step	Action
1	To see all volume options for both volumes, enter the following commands: **vol status -v *sec_vol*** **vol options *sec_vol*** **vol status -v *altvol*** **vol options *altvol***

Step	Action
2	Compare the output to ensure that all options, except for size-related options, are set to the same values. **Note** It is especially important that the language settings be the same.

Restart all SnapVault relationships in the new volume: The qtrees on the new volume will all be in the `normal` state, not the `snapvaulted` state (as shown using the `qtree status` command). You must restart the SnapVault relationships in the new volume.

Note

To support data integrity in a restrictive SnapLock Compliance environment, the `snapvault start -r` operation saves all data that was written after the common Snapshot copy to a directory on the volume. For details see "SnapVault resynchronization for SnapLock Compliance volumes" on page 298.

To restart SnapVault relationships in the new volume, complete the following steps.

Step	Action
1	To see the relationships in the old volume, enter the following command: **`snapvault status`** You will see relationships for all volumes. The relationships that are listed can be used to generate the relationships in the new volume.

Step	Action
2	To restart SnapVault relationships, enter the following command for each secondary qtree: **snapvault start -r -S *pri_system*:/vol/*pri_vol*/*pri_qtree* /vol/*altvol*/*dest_qtree*** See the na_snapvault(1) man page for details about the `snapvault start` command. **Note** If you leave out the `-r` option from the `snapvault start` command, the restart will fail.

Reconfigure the SnapVault schedules in the new volume: Assuming you are not changing the SnapVault Snapshot schedule on the primary, use the same names and schedule for Snapshot copies.

To reconfigure the SnapVault schedules, complete the following step.

Step	Action
1	Enter the following command: **snapvault snap sched -x -o retention_period=period *altvol snapname sched_spec*** See the na_snapvault(1) man page for details about the `snapvault snap sched` command. **Note** Specifying the retention period is optional. If a retention period is not specified, the default retention period for the volume is used.

Run a few snapshot targets in the new volume to ensure everything was migrated successfully: After the first scheduled Snapshot copy and transfer have occurred, look at the entries in the new volume.

To view the entries, complete the following steps.

Step	Action
1	Enter the following commands: **snapvault status** **snapvault status -c**
2	Look at all the entries for the new volume and ensure that everything is working correctly.

Stop all SnapVault relationships in the old volume: When everything is working correctly, you can delete the old relationships. To delete the old relationships, complete the following steps.

Step	Action
1	From the system that contains the old volume, enter the following command for all the qtrees in the old volume: **snapvault stop -f /vol/*sec_vol*/*dest_qtree***
2	From the primary, enter the following command for all the qtrees backed up to the old volume: **snapvault release /vol/*pri_vol*/*pri_qtree*** ***sec_system*:/vol/*sec_vol*/*dest_qtree*** See the na_snapvault(1) man page for details about the `snapvault release` command.

The `snapvault stop` command deletes the qtree from the active file system on the old volume. Backup copies of the qtree are in the Snapshot copies on the old volume and you can access them if you want to browse the old backups or restore from the old backups. The active image of the backup data is moved to the new volume when you restart the SnapVault relationship using the `snapvault start -r` command.

Managing SnapVault log files

Regulatory compliance of SnapVault log files

SnapVault creates log files that meet the following requirements for regulatory compliance:

- The contents of log files cannot be overwritten or deleted, nor can the log file itself be deleted.
- The log files must accurately record the exact series of events that occur during a timeframe.

How SnapVault maintains compliance

SnapVault uses a SnapLock volume in which the log files are kept. This volume, called a LockVault™ log volume, is a standard WORM volume. SnapVault uses an append-only write to write to the log files. This allows accurate record keeping but does not allow previous events to be overwritten.

SnapVault uses two types of log files to record events: SnapVault operations log files and SnapVault files transferred log files.

Configuring the LockVault log volume

The LockVault log volume must be created and configured before you begin an initial (level 0) SnapVault transfer. A LockVault log volume must be a SnapLock volume. The SnapLock volume can be a traditional volume or a flexible volume and is created using the `vol -L` or `aggr -L` command followed by a `vol create` command, respectively. See the SnapLock section of the *Storage Management Guide* for information about creating SnapLock volumes. All SnapVault log entries will be created in this log volume. This is a system-wide option; therefore, a SnapVault operation on any SnapLock volume on the system is logged to this volume.

Note

You need to set this option for SnapVault transfers to SnapLock volumes to succeed.

To configure the log volume, complete the following step on the SnapVault secondary system:

Step	Action
1	Enter the following command: **`options snapvault.lockvault_log_volume volume_name`** *volume_name* is a SnapLock volume. **Note** You must use the name of a previously created SnapLock volume for this command to succeed and this volume should not be used for any other purpose.

Where log files are kept

All SnapVault log files are kept in the /etc/log directory of the LockVault log volume that you created and configured. Log files in this directory inherit the default retention period of the log volume. Log files cannot be deleted from the volume until their retention periods have expired. SnapVault does not remove expired log files automatically; therefore, you will have to delete them.

Caution
Ensure that you set the default retention period on the log volume appropriately. The initial default retention period is 30 years for SnapLock Compliance volumes.

Operations log files: The following is the format for an operations log file name:

`snapvault.yyyymmdd_zzz`

yyyy is the year

mm is the month (01 to 12)

dd is the date (01 to 31)

zzz is the current time zone setting

The timestamp denotes when the log file is created and is generated using the compliance clock.

Operations log files have a weekly rotation policy; SnapVault creates a new log file every Sunday morning at midnight and is based on the compliance clock. Any SnapVault operations that are active when the new log file is created are recorded in the new log file.

Files transferred log files: SnapVault creates a new files transferred log file at the beginning of each SnapVault transfer. Files transferred log files are kept in the following directory structure in the LockVault log volume:

/etc/logs/snapvault secondary vol/secondary qtree name/month/log file

The snapvault secondary vol directories are the SnapLock volume names on the secondary storage system. The following is the format of the directory name:

secondary volume name_volume id

The volume UUID globally uniquely identifies a volume.

The secondary qtree name directories are the qtrees within a snapvault secondary vol directory. The following is the format of the secondary qtree directory name:

secondary qtree name_treeid_inodenum_gennum

The tree ID, inode number, and generation number uniquely identify the qtrees within a volume.

The month directory denotes the month in which log files for a qtree were generated. The format for the month directory is as follows:

yyyymm

The year (*yyyy*) and month (*mm*) are generated using the compliance clock.

The log file name uniquely identifies each transfer log file. The following is the format of the transfer log file name:

snapvault_filelog.*yyyymmdd_hhmmss_zzz*

The year (*yyyy*), month (*mm*), day (*dd*), hour (*hh*), minute (*mm*), second (*ss*) and time zone (*zzz*) are generated using the compliance clock.

Example: The following is a sample path for a transfer log file named snapvault_filelog.20040123_000500_PST:

```
/etc/log/vault_ ed0ad520-5f40-11d8-91ca-00a09800dcba/users_
19_1435628_1286197/200401/snapvault_filelog.20040123_000500_PST
```

Files transferred log file contents

A transfer log contains a header that describes the transfer and zero or more entries that describe the contents of the transfer. A log file contains only a header and zero entries in two instances: When no data on the primary had changed since the last SnapVault update, or for a rollback transfer.

Header format: The header at the beginning of each log file contains the following information:

- Transfer type
- Source system and source path
- Destination system and destination path
- Date and time of the start of the transfer
- Date and time of the source system Snapshot copy

Note

Date and time of the source system Snapshot copy is interpreted according to the time zone settings on the secondary system, not the primary system.

Example: The following is an example of a header log record:

```
# Transfer type: Base Start
# From: sourcesystem:/vol/compat_data/myqtree1
# To: destinationsystem:/vol/logs/mult1
# Start time: Tue Mar 30 22:43:09 GMT 2004
# Source snapshot timestamp: Wed Mar 31 23:41:01 EST 2004
```

Types of log entries: The following types of log entries are recorded in transfer log files:

- Create file (regular, special, or stream)
- Delete file (regular, special, or stream)
- Create directory
- Delete directory
- Rename from (regular file or directory)
- Rename to (regular file or directory)
- Modify file (regular, special, or stream file, but not directory)
- Modify attributes (regular, special, or stream file, or directory)

Log entry format: Log entries have the following format:

date_and_time action_taken base_path_length:stream_name_length path_name

date_and_time is the mtime or ctime value from the primary storage system, not the secondary storage system.

Note

The mtime value is used for Create file and directory types, Delete file and directory types, and Modify File type because these entry types modify data. The ctime value is used for Rename file and directory types and Modify Attribute types because these types modify the container for the data, not the data itself.

action_taken is one of the types of log entries.

base_path_length:stream_name_length is the length of the base path followed by a colon and the stream name in single byte characters. Only the base path length is shown if the file is not a stream file.

path_name is the file path relative to the base of the tree that is mirrored, not the absolute path name. The relative path name is the same on the primary and secondary storage systems.

Example: The following are examples of log entries:

```
Fri Mar 26 23:08:33 GMT 2004 Create Dir    7   ./mydir
Mon Oct  9 17:36:14 GMT 2000 Create File  14   ./mydir/myfile
Mon Jun 12 22:21:11 GMT 2000 Create File  14:8 ./mydir/myfile:strm
```

What actions create log entries: The following table lists actions that cause SnapVault to create a log entry, the type of log entry created, and possibly a note about the action.

Action	Entry type	Note
The initial transfer of all files and directories.	Create File Create Directory	
The data in a file was modified.	Modify File	A Modify Attributes entry is not created if a Modify File entry was.
The attributes of a directory, such as permissions, were changed or any entries in the directory were added or deleted.	Modify Attributes (directory)	
An access control list (ACL) for a file or directory was created, deleted, or changed.	Modify Attributes (any file or directory associated with the ACL)	Creation, deletion, or modification of ACLs are not explicitly logged.

Action	Entry type	Note
A file or directory with only one link was renamed.	Rename From Rename To	Renaming creates two entries that appear together in the log.
A file was renamed.	Modify Attributes or Modify File in addition to Rename From and Rename To	A Modify Attributes entry is created if no data was modified. A Modify file entry is created if data was modified.
A directory was renamed.	Modify Attributes in addition to Rename From and Rename To	
A hard link was added to an existing file.	Create File	
A hard link to a file was removed and the file still has one or more hard links.	Delete File	
A hard link was added to a file, but no content was changed.	Modify Attributes for all links to the file except the new link	
A hard link was added to a file and content was changed.	Modify File for all links to the file except the new link	
A hard link was deleted from a file, but no content was changed.	Modify Attributes for all links to the file except the deleted link	
A hard link was deleted from a file and content was changed.	Modify File for all links to the file except the deleted link	
A file with one or more hard links to it was created between the previous transfer and the current transfer.	Create File	No Modify File or Modify Attributes entry for the file
A file and all its hard links were deleted between the previous transfer and the current transfer.	Delete File	No Modify File or Modify Attributes entry for any links

Action	Entry type	Note
A file with multiple hard links was renamed.	Create File Delete File Modify Attributes or Modify File	A file with hard links is renamed by adding one link to the file and deleting one link from the file.
Data or attributes of a file were modified.	Modify File or Modify Attributes respectively	A Modify File or Modify Attributes entry is created for each link to the file except links that were added or deleted by the transfer.

Protecting your Compliant SnapVault backups with SnapMirror

Protection of different SnapLock volumes

If you want to protect backups on SnapLock Enterprise volumes with SnapMirror, you can use the same procedures that you use for non-SnapLock volumes. See "Protecting your SnapVault backups through SnapMirror" on page 340.

This section describes protecting backups on SnapLock Compliance volumes. If you are protecting backups on SnapLock Compliance volume with SnapMirror, some operations described in "Protecting your SnapVault backups through SnapMirror" on page 340 do not work. Specifically, the `snapmirror resync` command cannot be used on SnapLock Compliance volumes.

Note

The `snapmirror resync` command is not supported for SnapLock Compliance volumes because a resynchronization operation might alter data and, by definition, Compliance data must not be altered. Qtree SnapMirror (using the snapmirror resync command) and SnapVault resynchronization (using the `snapvault start -r` command) to SnapLock Compliance volumes are allowed because Data ONTAP provides a mechanism to store the data altered by a resynchronization for subsequent retrieval. No such mechanism exists for the `snapmirror resync` command.

Providing backup and standby protection

By setting up a SnapMirror relationship between the SnapVault secondary storage system and a SnapMirror destination or NearStore system, you can provide backup and standby protection for the SnapVault secondary storage system data. Backup and standby protection involves the following process:

Step	Action
1	Use 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 "Using backup and standby protection for SnapVault" on page 335. **Note** The SnapMirror destination systems cannot take advantage of the disk-space-saving benefits of volume cloning. If you used volume cloning on the original SnapVault secondary volume to store more than 255 Snapshot copies, the limit imposed on volumes by Data ONTAP, the SnapMirror destination volumes might need to be larger than the SnapVault secondary volumes if they are going to store the same amount of data.
2	Reestablishing backup and standby protection using this procedure, you have your original SnapMirror destination volumes as your SnapVault secondary volumes, and new volumes as your SnapMirror destination volumes. For details see "Reestablishing backup and standby protection for SnapVault" on page 336. **Note** You could use the original SnapVault secondary volumes when they become available after the retention periods of Snapshot copies and files on the volumes expire, but retention periods are usually long enough such that the volumes are not likely to be available.

Step	Action
3	An optional step, but one we do not recommend, is to return to the original backup and standby configuration. After reestablishing the backup and standby configuration your SnapVault secondary volumes are protected with a SnapMirror replication, but the two backup systems are reversed (the system that was your standby is now your secondary and the system that was your secondary is now your standby). Instead of returning to the original configuration, we recommend that the storage systems you use for your SnapVault secondaries and their associated SnapMirror standbys have the same configuration such that their roles can be reversed, if necessary. If you must return to the original backup and standby configuration, see “Returning to the original backup and standby configuration” on page 337 for details.

Using backup and standby protection for SnapVault : To set up SnapMirror backup and standby protection for the SnapVault secondary storage system and fail over to the standby system, if needed, complete the following steps.

Step	Action
1	Use the `license` command to confirm that the SnapVault secondary storage device has both SnapVault secondary storage and 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 disk-based destination device (another storage or NearStore system). For a quick description of setting up a SnapMirror relationship with a disk-based destination device, see “Setting up a basic SnapMirror operation” on page 105.

Step	Action
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 systems. For details on converting a SnapMirror destination to a writable volume, see "Converting a destination to a writable volume or qtree" on page 168.
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.

Reestablishing backup and standby protection for SnapVault: To reestablish SnapMirror backup and standby protection for the new SnapVault secondary storage system, complete the following step.

Step	Action
1	Perform an initial SnapMirror transfer from the new SnapVault secondary volume to the new volume using the `snapmirror initialize` command. See "Setting up a basic SnapMirror operation" on page 105 for details. **Caution** Using the `snapmirror initialize` command will destroy all existing data on the new volumes. **Note** If you are able to use your original SnapVault secondary volumes as the new SnapMirror destinations and you used volume cloning to go beyond 255 Snapshot copies, which is the limit imposed on a volume by Data ONTAP, the original SnapVault secondary volumes might not be large enough to accommodate the replication. If this is the case, you can add disk space to the original SnapVault secondary volumes.

Returning to the original backup and standby configuration: To return to the original configuration, complete the following steps.

Step	Action
1	Break the SnapMirror relationship.
2	Follow Steps 4 and 5 of the procedure described in "Using backup and standby protection for SnapVault" on page 335 to fail over to the original SnapVault secondary.
3	Follow the procedure described in "Reestablishing backup and standby protection for SnapVault" on page 336. **Note** Another set of disks must be used for the standby SnapMirror destinations when you reestablish the SnapMirror relationships, unless the retention periods have expired on the disks.

Limitations to Compliance backup and standby service

The following are limitations to the backup and standby service:

- Once you have failed over to the standby device and have begun to take backups, you cannot reuse the original SnapVault secondary storage disks for protecting the new SnapVault secondary volumes until the retention periods for all Snapshot copies and data on the original SnapVault secondary storage disks have expired.

  For all practical purposes, the original SnapVault secondary storage disks cannot be used to protect the new SnapVault secondary volumes and cannot be used to resume their original role as SnapVault secondaries because typical retention periods are on the order of months to years.
- You cannot restore backup data to SnapVault secondary storage if the secondary storage uses SnapLock Compliance volumes, because of the constraints put on SnapLock Compliance volumes to eliminate the possibility of losing Compliance data.

Note

Failing over to a standby device is complex, has the aforementioned limitations, and is costly in terms of time and disk space. Failing over should only be done in the event of a permanent loss of the original SnapVault secondary storage in when the quick resumption of backups is critical enough to warrant the cost of reestablishing the original configuration.

Backup protection of LockVault log volumes

Along with backing up Compliance SnapLock secondary storage volumes, you might want to provide additional protection for LockVault log volumes.

Protecting a LockVault log volume: The LockVault log volume can be protected by creating a SnapMirror backup with the same update schedule that you have for other SnapMirror relationships between the SnapVault secondary storage system and the SnapMirror destination. To protect a LockVault log volume, complete the following step:

Step	Action
1	Create a SnapMirror relationship between the LockVault log volume and a new LockVault log volume on the new standby system. See "Setting up a basic SnapMirror operation" on page 105 for details.

Failing over to the standby system: If you convert the standby system to a Snapvault secondary system, you need to configure the LockVault log volume to continue to collect log information. To configure the LockVault log volume, complete the following steps.

Step	Action
1	Quiesce the SnapMirror relationship to the LockVault log volume on the converted standby system using the `snapmirror quiesce` command. **`snapmirror quiesce` *`volume_name`***
2	Break the SnapMirror relationship to the LockVault log volume on the converted standby system using the `snapmirror break` command. **`snapmirror break` *`volume_name`***
3	Configure the LockVault log volume on the converted standby system using the `options` command. **`options snapvault.lockvault_log_volume` *`volume_name`***

Reestablishing standby protection: If you reestablish standby protection for new SnapVault secondary volumes, you should also reestablish new standby protection for the LockVault log volume. As is the case with reestablishing

standby protection for SnapVault secondary volumes, you need a new or scratch volume for the new standby LockVault log volume. To reestablish standby protection, complete the following step.

Step	Action
1	Create a SnapMirror relationship between the LockVault log volume and a new LockVault log volume on the new standby system. See “Setting up a basic SnapMirror operation” on page 105 for details.

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 storage system, NearStore system, 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 uses 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 341.
- SnapMirror backup and restore protection for SnapVault uses 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 341.

Note

These services either do not apply or are implemented differently when protecting SnapLock Compliance volumes. Do not use the procedures in this section if you want to protect SnapLock Compliance SnapVault secondary storage. See "Protecting your Compliant SnapVault backups with SnapMirror" on page 333.

The process of using SnapMirror with SnapVault: The SnapVault secondary storage system will carry out SnapVault operations on its sources as usual. Then on a scheduled per-volume basis, the system will replicate the SnapVault 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 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 disk-based destination device (another storage system or NearStore).

   For a quick description of setting up a SnapMirror relationship with a disk-based destination device, see "Setting up a basic SnapMirror operation" on page 105.
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 systems.

   For details on converting a SnapMirror destination to a writable volume, see "Converting a destination to a writable volume or qtree" on page 168.
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 Data ONTAP storage system, 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 105.
   - For a description of setting up a SnapMirror relationship with a destination tape device, see "SnapVault-to-tape backup scenario" on page 203.
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 180.
   - 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 205.

Turning SnapVault off

Why you turn SnapVault off

You might want to turn SnapVault off because the files there are no longer important or current or have been moved to another location. When you no longer want SnapVault to run, you can turn it off using the `options snapvault.enable off` command.

How to turn SnapVault off

To turn off SnapVault on a 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.

VERITAS NetBackup and SnapVault

Using NetBackup to manage SnapVault relationships

VERITAS NetBackup (NBU) provides an alternative method of configuring and managing SnapVault backups of storage system data. SnapVault, together with NetBackup, provides an integrated backup and recovery solution that offers centralized management, and wizard-based administration for controlling SnapVault qtree imaging from a NetBackup management station.

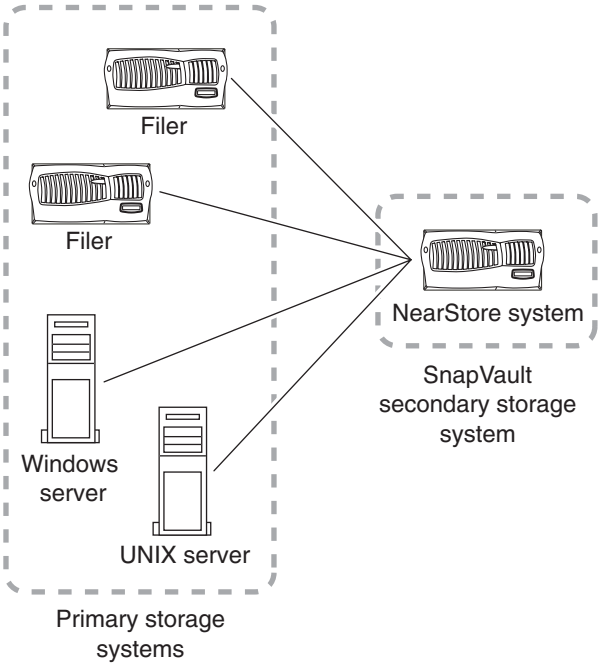
- An integrated solution—New and existing SnapVault relationships are controlled using the NetBackup Administration Console.
  - NetBackup takes over managing existing SnapVaulted qtrees
  - NetBackup creates new SnapVault relationships and manages subsequent backup schedules
- Centralized management—The NetBackup Administration Console handles most storage system interactions, eliminating the need to use CLI-based commands.
- Administrative control—SnapVault backups, Snapshot scheduling, and data recovery are intuitively managed using the NetBackup graphical user interface (GUI).

NBU SnapVault

NBU SnapVault software combines SnapVault and NetBackup functionality to support and manage backup schedules, and data recovery using the NetBackup Administration Console.

To support this system configuration, the storage system primary and secondarys are set up to allow administrative control from a NetBackup management station. When a data transfer is initiated from the NetBackup Administration Console, NetBackup locates the SnapVault primary storage system, and takes a Snapshot copy. The NetBackup server then contacts the secondary storage system, and

initiates a SnapVault backup of a primary system qtree. NBU SnapVault software enables NetBackup to manage SnapVault operations on both the primary and secondary storage systems.



For detailed information

The following sections discuss NetBackup management issues:

- "Setting up for NetBackup management" on page 346
- "Using NetBackup to manage SnapVault qtrees" on page 350
- "Scheduling efficient backup policies" on page 352

Setting up for NetBackup management

Requirements to support NetBackup administration

To set up SnapVault support for NetBackup transfers to storage systems, your configuration must meet the following requirements:

- The primary storage systems must be running Data ONTAP software version 6.5.1 or later.
- The secondary storage systems must be running Data ONTAP software version 7.1 or later.
- Snapvault must be licensed and enabled on the primary and secondary storage systems.
- NDMP service must be enabled on the primary and secondary storage systems as well as the NetBackup management station.
- The VERITAS NetBackup management station must be running NetBackup software version 6.0 or later.

What you need to do to enable SnapVault with NetBackup

You need to complete the following tasks to activate a SnapVault NetBackup relationship:

- If SnapVault is not already licensed on the primary and secondary storage systems, enter the appropriate license on each system.
- Set `options snapvault.enable` on for both the primary and secondary storage systems to enable SnapVault data transfers and Snapshot creation.
- Enable the NDMP service, `ndmpd on`, on both the primary and secondary storage systems.
- Set up `options snapvault.access` to allow the primary and secondary storage systems to exchange data.
  - Setting this option on the SnapVault primary storage system determines which secondary storage systems can access data from that primary storage system.
  - Setting this option on the SnapVault secondary storage system determines which SnapVault primary storage systems can access the secondary storage system.
- Set up `options ndmpd.access` to allow NetBackup access to both the primary and secondary storage systems.

Configuring the primary and secondary storage systems

On each primary and secondary storage system log in to that system's console and complete the following steps:

Step	Description
1	Set up the SnapVault primary and secondary licenses on each storage system. Enter the following commands: **pri_system> license add sv_primary_license** **sec_system> license add sv_secondary_license**
2	Enable the NDMP service on each storage system. Enter the following command: **ndmpd on**
3	Add a new backup user to the Backup Operators useradmin group list. Enter the following command: **useradmin user add backupuser -g "Backup Operators"**
4	Generate an NDMP password for the new user. Enter the following command: **ndmpd password backupuser** The NDMP password is used to authenticate the NetBackup media server to the storage system. This password is required for setup on the NetBackup management station. **Note** It is important to avoid using `root` as an NDMP password because `root` is not encrypted, and could compromise the integrity of your storage system.
5	Enable SnapVault on each storage system. Enter the following command: **options snapvault.enable on**

Step	Description
6	Use the `options snapvault.access` command to allow access between both the primary and secondary storage systems. On each primary system, specify the name of the secondary host. **`options snapvault.access host=secondary`** On each secondary system, specify the name of the primary host(s). **`options snapvault.access host=primary1,primary2,primary3`**
7	To specify the name of the NetBackup management station on both the primary and secondary storage systems, enter the following command: **`options ndmpd.access host=`*`netbackup_server1`*** **Note** The primary storage system must be able to resolve the host name to an IP address in the /etc/hosts file, or else the system needs to be running DNS or NIS. You can also 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.

Disabling SnapVault schedules

Existing SnapVaulted qtrees are designated as NBU SnapVault when NetBackup establishes a relationship with the storage system. This does not modify or interrupt SnapVault Snapshot scheduling, and might continue to increment the total number Snapshot copies per volume. To control the number of data transfer operations, SnapVault Snapshot schedules should be disabled.

To prevent SnapVault from attempting to update NetBackup managed qtrees, use the `snapvault snap unsched` command to disable incremental backups for all NBU SnapVault volumes.

To display the currently configured Snapshot schedule, enter the `snapvault snap sched` command and view the basenames of all Snapshot sets for all SnapVault volumes on the storage system.

Example: `system> snapvault snap sched`

To turn off existing Snapshot schedules for the SnapVault secondary storage system, enter the following command:

```
snapvault snap unsched <volume_name> <snap_name>
```

Note

This command does not end the SnapVault relationships between the secondary qtrees and their primary source qtrees.

Checking the status of a transfer

To check the status of a data transfer use the `snapvault status` command to see how recently a qtree has been updated. Additionally, you can use the `snap list` command to view SnapVault qtree relationships involved in the most recent transfer from a primary qtree to a secondary qtree. This display can be used 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.

The NetBackup catalog

The NetBackup catalog is an internal database that resides on the NetBackup management station. This database contains a record of all backup activity and information about the media and storage devices. The NetBackup catalog can be used to monitor and view SnapVault qtree archives.

For more information, see the *VERITAS NetBackup System Administrator's Guide*.

Using NetBackup to manage SnapVault qtrees

Planning NBU SnapVault qtree locations

Planning your NBU SnapVault primary system qtrees or directories and their corresponding secondary system qtrees is helpful. Multiple primary system qtrees from multiple volumes can all be backed up to corresponding secondary system qtrees in a single NBU volume. The maximum number of secondary system Snapshot copies per volume is 255.

Primary system qtree or directory location example	Corresponding Secondary system qtree location example
systemA:/vol/vol1/qtreeAA	sv_secondary:/vol/sv_vol/qtreeAA
systemA:/vol/vol1/qtreeAB	sv_secondary:/vol/sv_vol/qtreeAB
systemB:/vol/vol1/qtreeBB	sv_secondary:/vol/sv_vol/qtreeBB
winsrvrA:K:\melzdir	sv_secondary:/vol/sv_vol/melzdir
ux_srvrB:/usrs/moz_acct	sv_secondary:/vol/sv_vol/moz_acct

Note

The `K:\melzdir` Windows server, and the UNIX server `/usrs/moz_acct` examples above represent client qtrees that reside on the primary storage system. The Windows server, `K:\melzdir` is a CIFS share where `K:` is a volume and `melzdir` is a qtree within that volume. The UNIX server, `/usrs/moz_acct`, refers to a qtree directly as a mount point, or a volume that is mounted on `/usrs` where `moz_acct` is a qtree within that volume.

All volume and qtree management is done using the VERITAS NetBackup Administration Console. See the *VERITAS NetBackup System Administrator's Guide* for information about using the Device Configuration Wizard to configure Data ONTAP storage systems.

NBU SnapVault policies for managing Snapshot copies

NBU SnapVault Snapshot copies are managed by Data ONTAP in response to requests from the NetBackup interface. Backup policies, however, are scheduled using the NetBackup Administration Console. Policies are a set of rules that

define backup criteria for a particular client or group of clients. For example, NetBackup policies control backup intervals, number of versions, and how long to retain a particular archive.

For more information, see the *VERITAS NetBackup System Administrator's Guide*.

Snapshot copies and qtrees

NBU SnapVault software stores secondary qtree updates in Snapshot copies on a destination volume. To avoid the 255 Snapshot copy maximum per volume, group policies with similar backup schedules, and limit the number of backups retained to approximately 250. The reason is that SnapVault creates temporary Snapshot copies during the update process. By maintaining a limit of 250 Snapshot copies on the volume, you ensure the temporary files can be created without interrupting routine backup processes.

Note

Policies are a set of rules that define backup criteria for a particular client or group of clients. For example, NetBackup policies control backup intervals, number of versions, and how long to retain a particular archive. For more information, see the *VERITAS NetBackup System Administrator's Guide.*

Example: Qtree */vol/src_vol/qtree1* on the primary system contains *file_a*, *file_b*, and *file_c*. If this qtree is backed up to */vol/dst_vol/dst_qtree1*, a destination qtree on a secondary storage system, a Snapshot copy of *dst_vol* on the secondary system would contain *dst_qtree1,* and *filea*, *fileb*, and *filec*. This Snapshot copy is appropriately named and appears in the following format:

SnapVaultBackup.<*volname*>.<*snapid*>.<*cpcount*>.*yyyymmdd_hhmmss_zzz*

snapid and *cpcount* are identifiers

yyyymmdd_hhmmss is the time at which the Snapshot copy was created, and *zzz* is timezone.

Note

Data ONTAP software shortens Snapshot volume names to 128 characters to ensure that a Snapshot copy names do not exceed the maximum 255 character limit.

All updates to the destination qtree content are stored as Snapshot images until they are deleted using NetBackup.

Scheduling efficient backup policies

Managing storage

The maximum number of Snapshot copies allowed per volume is 255, so it is necessary to create backup policies that optimize space on the volume, and avoid the 255 Snapshot copy limit.

Optimize Snapshot copy content

You can conserve space by backing up distinct primary qtrees from multiple storage systems to the same secondary volume at the same time.

Example: Assume that multiple primary storage systems contain qtree1 through qtree5. A backup policy specifies a sequential back up schedule for each qtree, and these qtrees are backed up to qtrees of the same name on the same secondary volume. Five backups are taken of each qtree, which results in five Snapshot copies. Assume these Snapshot copies are named snapshot1 through snapshot5 respectively, and each contains a backup of qtree1, qtree2, qtree3, qtree4, and qtree5.

Snapshot copy*	Contains...
snapshot1	qtree1, qtree2, qtree3, qtree4, qtree5
snapshot2	qtree1, qtree2, qtree3, qtree4, qtree5
snapshot3	qtree1, qtree2, qtree3, qtree4, qtree5
snapshot4	qtree1, qtree2, qtree3, qtree4, qtree5
snapshot5	qtree1, qtree2, qtree3, qtree4, qtree5

This policy conserves space because multiple qtrees are backed up at the same time. It is less desirable to set policies that back up single qtrees multiple times because doing so results in more Snapshot copies than necessary.

Maximize the number of qtree backups

You can increase Snapshot storage capacity by using FlexVol volumes. FlexVol volumes do not increase the amount of physical storage space, rather they share space within an aggregate with other FlexVol volumes, and each volume supports 255 Snapshot copies.

Example: Assume that qtree1, qtree2, qtree3, qtree4, qtree5, and qtree6 are backed up to the same volume. Qtree1, qtree2, and qtree3 are backed up twice a day, and qtree4, qtree5, and qtree6 are backed up once a day. After five days, there will be a total of ten Snapshot copies on the destination volume.

Interval	Snapshot copy*	Contains...
Day 1	snapshot1	qtree1, qtree2, qtree3, qtree4, qtree5, qtree6
	snapshot2	qtree1, qtree2, qtree3
Day 2	snapshot3	qtree1, qtree2, qtree3, qtree4, qtree5, qtree6
	snapshot4	qtree1, qtree2, qtree3
Day 3	snapshot5	qtree1, qtree2, qtree3, qtree4, qtree5, qtree6
	snapshot6	qtree1, qtree2, qtree3
Day 4	snapshot7	qtree1, qtree2, qtree3, qtree4, qtree5, qtree6
	snapshot8	qtree1, qtree2, qtree3
Day 5	snapshot9	qtree1, qtree2, qtree3, qtree4, qtree5, qtree6
	snapshot10	qtree1, qtree2, qtree3

At the 255 maximum Snapshot copy threshold for the volume, the number of backups stored equals 250 that contain qtree1 through qtree3, and 125 that contain qtree4 through qtree6. While effective, this policy could be more efficient if the data was backed up to multiple FlexVol volumes.

FlexVol volumes are managed separately within the physical resources of your storage system. By increasing the number of volumes within a fixed aggregate, you increase Snapshot storage capacity because each volume maintains a 255 Snapshot copy limitation. FlexVol volumes are extremely efficient for grouping primary qtrees with similar backup schedules.

Example: Assume two FlexVol volumes totaling the same amount of disk space. Qtree1, qtree2, and qtree3 are backed up to one FlexVol volume, and qtree4, qtree5, and qtree6 are backed up to the second FlexVol volume. In this case, the number of backups stored on each FlexVol volume equals 250 with one volume containing qtree1 through qtree3, and the other containing qtree4 through qtree6. The amount of space consumed is the same, but Snapshot storage capacity is optimized.

VERITAS NetBackup and NearStore

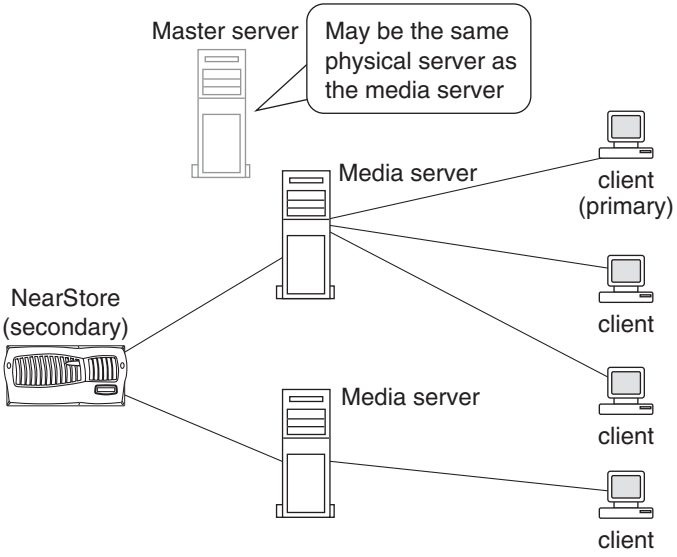
Using NetBackup (NBU) to manage data transfers to NearStore

NBU NearStore software enables data stored on client systems to be backed up to a NearStore storage system. A NearStore storage system, together with VERITAS NetBackup, provides an efficient backup and recovery solution for managing data across multiple platforms.

- An integrated solution—Data transfers are controlled using the NetBackup Administration Console.
  - NetBackup handles most storage system interactions, eliminating the need to use CLI-based commands.
  - Backup scheduling, and data recovery are intuitively managed using the NetBackup graphical user interface (GUI).
- Single instance storage—Backup file updates do not consume disk space unless blocks in the new backup image differ from blocks in the active file system.

About NBU NearStore

NBU NearStore software allows a NetBackup media server to backup data from multiple platforms, or clients, to a NearStore that is configured as a secondary storage system. These clients include: AIX, Solaris, HP-UX, Linux, Windows NT server, Windows 2000 server, or Windows 2003 server systems.



NetBackup data transfer

The NetBackup master or media server controls the flow of data from the primary storage system (client) to the secondary storage system (NearStore). Client data on the primary storage system is backed up inside a secondary qtree on the Nearstore.

Each data transfer to the NearStore secondary storage system consists of two data streams from the NetBackup media server.

- A standard NetBackup tar-style stream
- A metadata stream that is common to both Data ONTAP and NetBackup

The tar-style stream is expanded on the NearStore to create an active file system.

Note

NBU NearStore in Data ONTAP 7.1 does not allow administrative access to the active file system. However, you can view the file system from the NetBackup Administration Console. For more information, see the *VERITAS NetBackup System Administrator's Guide.*

No data block duplication

NearStore avoids duplicating disk blocks by comparing blocks in the most recent backup with the active file system. Incremental updates do not consume disk space unless blocks in the new backup image differ from blocks in the active file system. As a result, multiple backups to the same qtree store only uncommon blocks, and blocks that are common continue to share space.

To view storage savings, use the `df -s` command:

```
system*> df -s
Filesystem               used      shared       saved        %saved
/vol/vol0/            1335000           0           0            0%
/vol/flexsle/              96           0           0            0%
/vol/sim/                 292           0           0            0%
/vol/p3/             21304124    14637384    21731976           50%
```

For detailed information

The following sections discuss ways to enable and manage data transfers to NearStore systems using NetBackup:

- "Setting up SnapVault support for NetBackup transfers" on page 357
- "Managing data transfers" on page 360

Setting up SnapVault support for NetBackup transfers

Requirements

To set up SnapVault support for NetBackup transfers to a NearStore storage system, your configuration must meet the following requirements:

- The storage system must be a NearStore.
- The NearStore system must be running Data ONTAP software version 7.1 or later.
- A SnapVault secondary license must be enabled on the NearStore system.
- NetBackup Enterprise 6.0 on the master server and media server
- NetBackup client systems must be running NetBackup 5.0 or later.
- NDMP service must be enabled on the NearStore system.
- An NBU volume must be a FlexVol volume.

Note

NBU volumes support only NetBackup data transfer and storage operations. Therefore, it is important to disable SnapVault schedules and release SnapVault relationships on NBU volumes. See, "Disabling SnapVault schedules" on page 358, and "Releasing SnapVault relationships" on page 359 for more information.

Configuring NearStore as a secondary storage system

To configure the NearStore as a secondary storage system. Complete the following steps.

Step	Description
1	Set up the SnapVault secondary license. Enter the following command: `license add sv_secondary_license`
2	Add a new backup user to the Backup Operators useradmin group list. Enter the following command: `useradmin user add backupuser -g "Backup Operators"`

Step	Description
3	Generate an NDMP password for the new user. Enter the following command: **`ndmpd password backupuser`** The NDMP password is used to authenticate the NetBackup media server to the NearStore storage system. This password is required for NearStore setup on the NetBackup management station. **Note** It is important to avoid using `root` as an NDMP password because `root` is not encrypted, and could compromise the integrity of your storage system.
4	Enable SnapVault. Enter the following command: **`options snapvault.enable on`**
5	Use the `options snapvault.access` command to specify the names of the NetBackup media servers. Enter the following command: **`options snapvault.access host=netbackup_server1, netbackup_server2 ...`** **Note** NetBackup client connect requests are received on TCP port 10571.

NBU NearStore and NBU SnapVault cannot share a volume

NBU NearStore software and NBU SnapVault functionality cannot share the same volume. To prepare a FlexVol volume to support NetBackup:

- Disable any incremental backups to the secondary qtrees that were originally scheduled for the FlexVol volume
- Release existing SnapVault relationships

Disabling SnapVault schedules

To turn off existing Snapshot schedules for the SnapVault secondary storage system, enter the following command:

```
snapvault snap unsched <volume_name> <snap_name>
```

Note

This command does not end the SnapVault relationships between the secondary system qtrees and their platform source drives or directories.

To disable Snapshot schedules on the volume:

```
snap sched <volume_name> 0 0 0
```

Releasing SnapVault relationships

To release SnapVault relationships and free the space on the volume, enter the following commands.

Step	Action
1	Enter one of the following commands. ◆ On the primary system console, enter: **snapvault release prim_qtree_path sec_system:sec_qtree_path** ◆ On the secondary system console, enter: **snapvault release sec_qtree_path prim_system:prim_qtree_path**

Example:

```
system> snapvault release /vol/sv_vol/qtree3
systemA:/vol/vol1/qtree3
```

Scheduling backups

All data transfers are managed by Data ONTAP in response to requests from the NetBackup interface. Backup policies, however, are scheduled using the NetBackup Administration Console. Policies are a set of rules that define backup criteria for a particular client or group of clients. For example, NetBackup policies control backup intervals, number of versions, and how long to retain a particular archive.

For more information, see the *VERITAS NetBackup System Administrator's Guide*.

File naming

Default file system naming for NBU NearStore qtree images is as follows:

`.nbu_image_data` is a special qtree that contains information specific to the tar images associated with each qtree.

`nbu_clientname_policyname_suffix` is the qtree naming convention where each name is specific to NetBackup client and policy settings.

Each qtree contains a `.nbu_no_access` subdirectory. NBU NearStore in Data ONTAP 7.1 does not support viewing the contents of this subdirectory from the storage system console. However, you can view the backup files from the NetBackup Administration Console.

For more information, see the *VERITAS NetBackup System Administrator's Guide.*

Monitoring qtree status

To confirm that a destination qtree has been created on the storage system, use the `qtree status` command to display NBU volume status.

```
system*> qtree status
Volume    Tree    Style  Oplocks   Status
--------- -------- ----- -------- ---------
vol0              unix   enabled   normal
flexsle           unix   enabled   normal
p3                unix   enabled   normal
p3       .nbu_image_data unix   enabled   snapvaulted
p3       nbu_clientname_1_daily_0000 unix  enabled  snapvaulted
```

```
p3       nbu_clientname_2_daily_0000 unix  enabled  snapvaulted
p3       nbu_clientname_daily_0001   unix  enabled  snapvaulted
p3       nbu_clientname_daily_0002   unix  enabled  snapvaulted
p3       nbu_clientname_daily_0003   unix  enabled  snapvaulted
```

Checking the status of a data transfer

To check the status of a data transfer, use the `snapvault status` command. This display can be used to view recent backup activity and confirm transfer status.

```
system*> snapvault status
Snapvault secondary is ON.
```

```
Source          Destination                                State         Lag       Status
system1:daily    client:/vol/p3/nbu_client_1_daily_0000  Snapvaulted   00:35:32   Idle
system2:daily    client:/vol/p3/nbu_client_2_daily_0000  Snapvaulted   00:35:32   Idle
system:daily     client:/vol/p3/nbu_client_daily_0001    Snapvaulted   00:35:32   Idle
system:daily     client:/vol/p3/nbu_client_daily_0002    Snapvaulted   00:35:32   Idle
system:daily     client:/vol/p3/nbu_client_daily_0003    Snapvaulted   00:35:32   Idle
```

SnapLock Storage Management 6

About this chapter

This chapter describes how to use SnapLock volumes and aggregates to provide WORM (write-once-read-many) storage.

Topics in this chapter

This chapter discusses the following topics:

- "About SnapLock" on page 364
- "Creating SnapLock volumes" on page 366
- "Managing the compliance clock" on page 368
- "Setting volume retention periods" on page 370
- "Destroying SnapLock volumes and aggregates" on page 373
- "Managing WORM data" on page 375

About SnapLock

What SnapLock is

SnapLock is an advanced storage solution that provides an alternative to traditional optical WORM (write-once-read-many) storage systems for non-rewritable data. SnapLock is a license-based, open-protocol feature that works with application software to administer non-rewritable storage of data.

SnapLock is available in two forms: SnapLock Compliance and SnapLock Enterprise.

SnapLock Compliance: Provides WORM protection of files while also restricting the storage administrator's ability to perform any operations that might modify or erase retained WORM records. SnapLock Compliance should be used in strictly regulated environments that require information to be retained for specified lengths of time, such as those governed by SEC Rule 17a-4.

SnapLock Enterprise: Provides WORM protection of files, but uses a trusted administrator model of operation that allows the storage administrator to manage the system with fewer restrictions. For example, SnapLock Enterprise allows the administrator to perform operations, such as destroying SnapLock volumes, that might result in the loss of data.

Note

SnapLock Enterprise should not be used in strictly regulated environments.

How SnapLock works

WORM data resides on SnapLock volumes that are administered much like regular (non-WORM) volumes. SnapLock volumes operate in WORM mode and support standard file system semantics. Data on a SnapLock volume can be created and committed to WORM state by transitioning the data from a writable state to a read-only state.

Marking a currently writable file as read-only on a SnapLock volume commits the data as WORM. This commit process prevents the file from being altered or deleted by applications, users, or administrators.

Data that is committed to WORM state on a SnapLock volume is immutable and cannot be deleted before its retention date. The only exceptions are empty directories, and files that are not committed to a WORM state. Additionally, once directories are created, they cannot be renamed.

In Data ONTAP 7.0 and later versions, WORM files can be deleted after their retention dates. The retention date on a WORM file is set when the file is committed to WORM state, but it can be extended at any time. The retention period can never be shortened for any WORM file.

Licensing SnapLock functionality

SnapLock functionality can be licensed as SnapLock Compliance or SnapLock Enterprise. These licenses are mutually exclusive, but can co-exist if enabled for different volumes and aggregates on a single storage system.

- SnapLock Compliance

  A SnapLock Compliance volume is recommended for strictly regulated environments. This license enables basic functionality and restricts administrative access to files.

- SnapLock Enterprise

  A SnapLock Enterprise volume is recommended for less regulated environments. This license enables general functionality, and allows you to store and administer secure data.

Note

FlexVol volumes created within a licensed aggregate will inherit the license attributes specific to that aggregate.

AutoSupport with SnapLock

If AutoSupport is enabled, the storage system sends AutoSupport messages to Technical Support. These messages include event and log-level descriptions. SnapLock volume state and options are included in AutoSupport output.

Replicating SnapLock volumes

You can replicate SnapLock volumes to another storage system using the SnapMirror feature of Data ONTAP. If an original volume becomes disabled, SnapMirror ensures quick restoration of data. For more information about SnapMirror and SnapLock, see the *Data Protection Online Backup and Recovery Guide*.

Creating SnapLock volumes

SnapLock is an attribute of the containing aggregate

Although this guide uses the term "SnapLock volume" to describe volumes that contain WORM data, in fact SnapLock is an attribute of the volume's containing aggregate. Because traditional volumes have a one-to-one relationship with their containing aggregate, you create traditional SnapLock volumes much as you would a standard traditional volume. To create SnapLock FlexVol volumes, you must first create a SnapLock aggregate. Every FlexVol created in that SnapLock aggregate is, by definition, a SnapLock volume.

Creating SnapLock traditional volumes

SnapLock traditional volumes are created in the same way a standard traditional volume is created, except that you use a `-L <license_attribute>` parameter with the `vol create` command. If a specific license is already enabled, there is no need to specify a `compliance` or `enterprise` attribute. However, when creating a volume, if both SnapLock Compliance and SnapLock Enterprise licenses are enabled, use the `compliance` or `enterprise` attribute to differentiate between licenses.

Verifying volume or aggregate status

You can use the `vol status` command to verify that the newly created SnapLock volume or aggregate exists. The following example displays `vol status` output, where the attribute of the SnapLock volume is in the Options column.

```
sys1> vol status

      Volume State      Status            Options
        vol0 online     raid4, trad       root
    wormvol online      raid4, trad       no_atime_update=on,
                                          snaplock_compliance
```

Creating SnapLock aggregates

SnapLock aggregates are created in the same way a standard aggregate is created, except that you use the `-L` parameter with the `aggr create` command. However, when creating an aggregate, if both SnapLock Compliance and SnapLock Enterprise licenses are enabled, use the `compliance` or `enterprise` attribute to differentiate between licenses.

SnapLock write_verify option

Data ONTAP provides a write verification option for SnapLock Compliance volumes: `snaplock.compliance.write_verify`. When this option is enabled, an immediate read verification occurs after every disk write, providing an additional level of data integrity.

Note

The SnapLock write verification option provides negligible benefit beyond the advanced, high-performance data protection and integrity features already provided by NVRAM, checksums, RAID scrubs, media scans, and double-parity RAID. SnapLock write verification should be used where the interpretation of regulations requires that each write to the disk media be immediately read back and verified for integrity.

SnapLock write verification comes at a performance cost and may affect data throughput on SnapLock Compliance volumes.

Managing the compliance clock

SnapLock Compliance requirements to enforce WORM retention

SnapLock Compliance meets the following requirements needed to enforce WORM data retention:

- Secure time base—ensures that retained data cannot be deleted prematurely by changing the regular clock of the storage system
- Synchronized time source—provides a time source for general operation that is synchronized to a common reference time used inside your data center

How SnapLock Compliance meets the requirements

SnapLock Compliance meets the requirements by using a secure compliance clock. The compliance clock is implemented in software and runs independently of the system clock. Although running independently, the compliance clock tracks the regular system clock and remains very accurate with respect to the system clock.

Initializing the compliance clock

To initialize the compliance clock, complete the following steps.

Caution

The compliance clock can be initialized *only once* for the system. You should exercise extreme care when setting the compliance clock to ensure that you set the compliance clock time correctly.

Step	Action
1	Ensure that the system time and time zone are set correctly.
2	Initialize the compliance clock using the following command: `date -c initialize` **Result:** The system prompts you to confirm the current local time and that you want to initialize the compliance clock.
3	Confirm that the system clock is correct and that you want to initialize the compliance clock.

Example: `system> date -c initialize`

```
*** WARNING: YOU ARE INITIALIZING THE SECURE COMPLIANCE CLOCK ***

You are about to initialize the secure Compliance Clock of this
system to the current value of the system clock. This procedure
can be performed ONLY ONCE on this system so you should ensure
that the system time is set correctly before proceeding.

The current local system time is: Wed Feb 4 23:38:58 GMT 2004

Is the current local system time correct? y
Are you REALLY sure you want initialize the Compliance Clock? y

Compliance Clock: Wed Feb 4 23:39:27 GMT 2004
```

Viewing the compliance clock time

To view the compliance clock time, complete the following step.

Step	Action

1	Enter the command:
	`date -c`
	Example:
	`date -c`
	`Compliance Clock: Wed Feb 4 23:42:39 GMT 2004`

Setting volume retention periods

When you should set the retention periods

You should set the retention periods after creating the SnapLock volume and before using the SnapLock volume. Setting the options at this time ensures that the SnapLock volume reflects your organization's established retention policy.

SnapLock volume retention periods

A SnapLock Compliance volume has three retention periods that you can set:

Minimum retention period: The minimum retention period applies to the shortest amount of time the WORM file must be kept in a SnapLock volume. You set this retention period to ensure that applications or users do not assign noncompliant retention periods to retained records in regulatory environments. This option has the following characteristics:

- Existing files that are already in the WORM state are not affected by changes in this volume retention period.
- The minimum retention period takes precedence over the default retention period.
- Until you explicitly reconfigure it, the minimum retention period is 0.

Maximum retention period: The maximum retention period applies to the largest amount of time the WORM file must be kept in a SnapLock volume. You set this retention period to ensure that applications or users do not assign excessive retention periods to retained records in regulatory environments. This option has the following characteristics:

- Existing files that are already in the WORM state are not affected by changes in this volume retention period.
- The maximum retention period takes precedence over the default retention period.
- Until you explicitly reconfigure it, the maximum retention period is 30 years.

Default retention period: The default retention period specifies the retention period assigned to any WORM file on the SnapLock Compliance volume that was not explicitly assigned a retention period. You set this retention period to ensure that a retention period is assigned to all WORM files on the volume, even if users or applications failed to assign a retention period.

Caution

For SnapLock Compliance volumes, the default retention period is equal to the maximum retention period of 30 years. If you do not change either the maximum retention period or the default retention period, for 30 years you will not be able to delete WORM files that received the default retention period.

Setting SnapLock volume retention periods

SnapLock volume retention periods can be specified in days, months, or years. Data ONTAP applies the retention period in a calendar correct method. That is, if a WORM file created on 1 February has a retention period of 1 month, the retention period will expire on 1 March.

Setting the minimum retention period: To set the SnapLock volume minimum retention period, complete the following step.

Step	Action
1	Enter the following command: **`vol options vol_name snaplock_minimum_period period`** *vol_name* is the SnapLock volume name. *period* is the retention period in days (d), months (m), or years (y). **Example:** The following command sets a minimum retention period of 6 months: `vol options wormvol1 snaplock_minimum_period 6m`

Setting the maximum retention period: To set the SnapLock volume maximum retention period, complete the following step.

Step	Action
1	Enter the following command: **`vol options vol_name snaplock_maximum_period period`** *vol_name* is the SnapLock volume name. *period* is the retention period in days (d), months (m), or years (y). **Example:** The following command sets a maximum retention period of 3 years: `vol options wormvol1 snaplock_maximum_period 3y`

Setting the default retention period: To set the SnapLock volume default retention period, complete the following step.

Step	Action
1	Enter the following command: **`vol options vol_name snaplock_default_period [period | min | max]`** *vol_name* is the SnapLock volume name. *period* is the retention period in days (d), months (m), or years (y). `min` is the retention period specified by the `snaplock_minimum_period` option. `max` is the retention period specified by the `snaplock_maximum_period` option. **Example:** The following command sets a default retention period equal to the minimum retention period: `vol options wormvol1 snaplock_default_period min`

Destroying SnapLock volumes and aggregates

When you can destroy SnapLock volumes

SnapLock Compliance volumes constantly track the retention information of all retained WORM files. Data ONTAP does not allow you to destroy any SnapLock volume that contains unexpired WORM content. Data ONTAP does allow you to destroy SnapLock Compliance volumes when all the WORM files have passed their retention dates, that is, expired.

Note

You can destroy SnapLock Enterprise volumes at any time.

When you can destroy SnapLock aggregates

You can destroy SnapLock Compliance aggregates only when they contain no volumes. The volumes contained by a SnapLock Compliance aggregate must be destroyed first.

Destroying SnapLock volumes

To destroy a SnapLock volume, complete the following steps.

Step	Action
1	Ensure that the volume contains no unexpired WORM data.
2	Enter the following command to offline the volume: **`vol offline vol_name`**
3	Enter the following command: **`vol destroy vol_name`**

If there are any unexpired WORM files in the SnapLock Compliance volume, Data ONTAP returns the following message:

```
vol destroy: Volume volname cannot be destroyed because it is a
SnapLock Compliance volume.
```

Destroying SnapLock aggregates

To destroy a SnapLock aggregate, complete the following steps.

Step	Action
1	Using the steps outlined in "Destroying SnapLock volumes" on page 373, destroy all volumes contained by the aggregate you want to destroy.
2	Take all FlexVol volumes offline and destroy them by entering the following commands for each volume: **`vol offline vol_name`** **`vol destroy vol_name`**
3	Take the aggregate offline by entering the following command: **`aggr offline aggr_name`** *aggr_name* is the name of the aggregate that you intend to destroy and whose disks you are converting to hot spares.
4	Destroy the aggregate by entering the following command: **`aggr destroy aggr_name`** *aggr_name* is the name of the aggregate that you intend to destroy and whose disks you are converting to hot spares.

Managing WORM data

Transitioning data to WORM state and setting the retention date

After you place a file into a SnapLock volume, you must explicitly commit it to a WORM state before it becomes WORM data. The last accessed timestamp of the file at the time it is committed to WORM state becomes its retention date.

This operation can be done interactively or programmatically. The exact command or program required depends on the file access protocol (CIFS, NFS, etc.) and client operating system you are using. Here is an example of how you would perform these operations using a Unix shell:

Unix shell example: The following commands could be used to commit the document.txt file to WORM state, with a retention date of November 21, 2004, using a Unix shell.

```
touch -a -t 200411210600 document.txt
chmod -w document.txt
```

Note

In order for a file to be committed to WORM state, it must make the transition from writable to read-only in the SnapLock volume. If you place a file that is already read-only into a SnapLock volume, it will not be committed to WORM state.

If you do not set the retention date, the retention date is calculated from the default retention period for the volume that contains the file.

Extending the retention date of a WORM file

You can extend the retention date of an existing WORM file by updating its last accessed timestamp. This operation can be done interactively or programmatically.

Note

The retention date of a WORM file can never be changed to earlier than its current setting.

Determining whether a file is in a WORM state

To determine whether a file is in WORM state, it is not enough to determine whether the file is read-only. This is because to be committed to WORM state, files must transition from writable to read-only *while in the SnapLock volume*.

If you want to determine whether a file is in WORM state, you can attempt to change the last accessed timestamp of the file to a date earlier than its current setting. If the file is in WORM state, this operation fails.

Volume Copy 7

About this chapter

This chapter discusses using the `vol copy` set of commands for replicating all of the data from one volume to another. 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 77.

Topics of this chapter

This chapter discusses the following topics:

- "Learning about volume copy" on page 378
- "Preparing to copy a volume" on page 380
- "Copying volumes" on page 388
- "Checking the status of a volume copy operation" on page 394
- "Displaying the current speed for copying a volume" on page 396
- "Controlling the speed of a volume copy operation" on page 397
- "Aborting a volume copy operation" on page 399

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 Snapshot copies from one volume to another. The source and destination volumes must be the same type (traditional or flex). 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 on the same or on a different storage system. The result 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 storage system 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 system. 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 Snapshot copies 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.

Maximum number of simultaneous volume copies

Volume copy has the same limit of simultaneous copies that SnapMirror replications have. See “Maximum number of simultaneous replication operations” on page 86.

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 storage system to another.	The destination storage system has more storage or is a model that supports newer technology.
You want to move a volume from one set of disks to another on the same storage system.	You want to ◆ Split a volume ◆ Expand storage **Examples:** You can copy the vol0 volume to the vol1 volume and then delete duplicate 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 storage system to another regularly to ensure high data availability.	After you copy the data, clients can switch to the destination storage system in the following scenarios: ◆ When you shut down the source system for software or hardware upgrades, or when the source system 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 system, clients can continue to have access to the files on the destination system while you are restoring the files to the source system. **Note** This is also a good application for SnapMirror.

Preparing to copy a volume

Requirements for copying a volume

The storage systems 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 source and destination volumes must be of the same type: either both traditional or both FlexVol volumes.
- The capacity of the destination volume must be greater than or equal to the capacity of the source volume.
- The source and destination storage systems 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 Snapshot copies 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 or on different storage systems.

Recommendation for copying a volume

When a storage system copies data between two volumes on separate systems, it floods the network between the two systems with packets. Users of the systems involved in a volume copy operation might notice a degradation in response time during the copy. A private network for copying between the source and destination storage systems helps circumvent network-related performance problems when copying to a different system.

For detailed information

The following sections discuss the ways you prepare for copying a volume:

- "Verifying the size of each volume" on page 382
- "Verifying the relationship between storage systems" on page 384
- "Verifying and changing status of source and destination volumes" on page 385
- "Enabling remote access" on page 387

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.

Verifying traditional volume size

To compare the file system size of traditional volumes, complete the following steps.

Step	Action
1	On the source system, enter the following command: **`vol status -b volume_name`** *volume_name* is the name of the source volume. **Result:** Data ONTAP displays the block size of the volume (in bytes), the RAID volume size, and the Write Anywhere File Layout (WAFL) file system size. If no volume name is given, information for all volumes is displayed.
2	On the destination system, 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 FlexVol volume size

You can compare the size of volumes using the `vol status -b` command as described above, or you can use the `vol size` command. It might be more convenient to use `vol size` command, because you use this command to change the size of a FlexVol volume.

To compare the file system size of FlexVol volumes, complete the following steps.

Step	Action
1	On the source system, enter the following command: **`vol size volume_name`** *volume_name* is the name of the source volume. **Result:** Data ONTAP displays the block size of the volume (in bytes), the RAID volume size, and the Write Anywhere File Layout (WAFL) file system size. If no volume name is given, information for all volumes is displayed.
2	On the destination system, repeat Step 1, replacing *volume_name* with the name of the destination volume.
3	Compare the sizes. If the size of the destination is the same as or larger than the size of the source, you can use the `vol copy` command (or SnapMirror) to transfer data from the source to the destination. If the destination volume is smaller than the size of the source, increase the size of the destination volume. For information on increasing the size of a volume see the volume management chapter in the *Storage Management Guide*.

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 storage systems

Creating trust relationships

If the source and destination volumes in a volume copy operation reside on two different storage systems, the systems must have a trust relationship with each other.

To specify each system as a trusted host of the other, complete the following steps.

Step	Action
1	By using FilerView or mounting the storage system with NFS, enter the destination system host name in the /etc/hosts.equiv file of the source system, 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 system allows that host to perform remote operations.
2	Repeat Step 1 on the destination system, entering the source system 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 system: **`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 storage system. 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 1 on the source system, 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 system: **`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 system: **`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:

```
systemA> 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 storage system, 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 394.

Number of vol copy operations supported

Whether Data ONTAP can execute a `vol copy start` command depends on how many volume copy operations are already in progress on the storage systems 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 a storage system, 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 system. Two of the operations are for reading the vol0 and vol2 volumes, and two of the operations are for writing the vol1 and vol3 volumes.

Example: Suppose you enter the following three `vol copy start` commands on a storage system named systemA to copy volumes to another storage system named systemB:

```
vol copy start vol0 systemB:vol0
vol copy start vol1 systemB:vol1
vol copy start vol2 systemB:vol2
```

When these commands are in progress, systemA runs three volume copy operations to read the volumes, and systemB runs three volume copy operations to write the volumes.

An additional `vol copy start` command to copy *between* systemA and systemB will succeed because the command adds one more volume copy operation to each storage system.

However, if you enter an additional `vol copy start` command to copy volumes *locally* on either systemA or systemB, it will fail. This is because the additional command creates two volume copy operations, one for reading and one for writing, on the storage system that performs the local copying.

Copying Snapshot copies with the vol copy start command

The following table describes the Snapshot copies that will be copied from the source volume and the resulting Snapshot copies on the destination volume, depending on the option you use with the `vol copy start` command.

Option	Snapshot copies to copy from the source volume	Snapshot copies in the Snapshot file system of the destination volume
None	No Snapshot copies are copied. Only the snap-shot taken after you enter the `vol copy start` command, are copied.	A Snapshot copy named snapshot_for_volcopy.*n* is created, where *n* is a number starting at 0 and incrementing by one whole number with each `vol copy` operation is created.

Option	Snapshot copies to copy from the source volume	Snapshot copies in the Snapshot file system of the destination volume
`-S`	All Snapshot copies in the Snapshot file system of the source volume, and the Snapshot copy taken after you enter the `vol copy start command,` are copied.	All Snapshot copy copies 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, are created.
`-s` followed by the name of the Snapshot copy	The specified Snapshot copy will be copied.	The specified Snapshot copy is created.

Note

The `vol copy start -S` command does not copy any Snapshot copies 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 copy named nightly.0 at midnight, Data ONTAP does not copy the nightly.0 Snapshot copy.

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 Snapshot copies to copy. *source_volume* and *dest_volume* are the names of the source and destination volumes. If a volume is on a different system, precede the volume name with the system 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 storage systems, you can enter the `vol copy start` command on either the source or destination system. You cannot, however, enter the command on a third system 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 Snapshot copies from the vol0 volume to the vol1 volume on the same storage system	`vol copy start -S vol0 vol1`
Copy a nightly Snapshot copy from the vol0 volume to the vol1 volume on the same storage system	`vol copy start -s nightly.1 vol0 vol1`
Create a Snapshot copy in the vol0 volume to be copied to the vol1 volume on the same storage system	`vol copy start vol0 vol1`

If you want to...	Use...
Copy all Snapshot copies from the vol0 volume to the vol1 volume on a different storage system named systemA	`vol copy start -S vol0 systemA:vol1`

Error messages generated by vol copy start commands

If your storage system 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 system systemB`
	The source system does not have permission to copy to the destination system.
	
	Action: Make sure that the storage systems 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.

Error message	Meaning
`write: setting up STDERR broken pipe`	A local volume copy tried to start, but Remote Shell access is not enabled on the system. **Action:** Enable Remote Shell access on the storage system 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 system 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 system prompt on the console when Data ONTAP is copying the volume.
- If data is being copied between two storage systems, you can enter the `vol copy status` command through a Remote Shell connection to either system. The operation numbers displayed on the source system and the destination system 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 storage system. When the operation is in progress, it displays the volume copy operation status.

```
systemA>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 storage system, as shown in the following example:

```
rsh systemA 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 slower 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 storage system.

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 systemA 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 storage systems, you can stop copying by executing the `vol copy abort` command on either system.

If you start the volume copying operation from the system console, you can enter the `vol copy abort` command only through the `rsh` command. This is because you do not have access to the system 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.

SyncMirror Management 8

About this chapter

This chapter describes how to use the optional SyncMirror feature of Data ONTAP to mirror aggregates embedded in traditional volumes and aggregates that contain FlexVol volumes. It also describes how to manage aggregates in mirrored relationships. For more information about aggregates and volumes, see the *Storage Management Guide*.

Topics in this chapter

This chapter discusses the following topics:

- "Understanding mirrored aggregates" on page 402
- "Enabling and disabling the mirroring license" on page 406
- "Creating mirrored aggregates and traditional volumes" on page 409
- "Adding disks to mirrored aggregate or traditional volumes" on page 423
- "Changing the state of a plex" on page 428
- "Splitting mirrored aggregates or traditional volumes" on page 430
- "Rejoining split aggregates or traditional volumes" on page 432
- "Removing and destroying a plex" on page 434
- "Comparing the plexes of mirrored aggregate or traditional volumes" on page 437

Understanding mirrored aggregates

About the SyncMirror feature

The SyncMirror software creates aggregates or traditional volumes that consist of two copies of the same WAFL file system. The two copies, known as plexes, are simultaneously updated; therefore, the copies are always identical.

Advantages of SyncMirror

A SyncMirror relationship between two aggregates or traditional volumes provides a high level of data availability because the two plexes are physically separated on different shelves and the shelves are connected to the storage system with separate cables and adapters. Each plex has its own collection of spare disks.

Physical separation of the plexes protects against data loss in the case of a double-disk error or loss of disk connectivity. 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 mirror relationship reestablished.

Another advantage of mirrored plexes is faster rebuild time.

In contrast, if a SnapMirrored aggregate or traditional 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 the last Snapshot copy was created (you must issue commands to make the partner's data available).

Disadvantage of SyncMirror

The disadvantage of SyncMirror is that a mirrored aggregate or traditional volume requires twice as many disks as an unmirrored aggregate or traditional 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 traditional volume—100 GB for each plex of the mirrored traditional volume, or 2,880 GB of disk space to mirror a 1,440-GB aggregate.

Prerequisites for using mirrored volumes

The following are prerequisites for using mirrored aggregates or traditional volumes:

- You must purchase and enable a SyncMirror license. You must enter this key to receive important product information prior to enabling the synchronous SnapMirror feature.

To enable the synchronous SnapMirror feature, use the `license add` command, and enter one of the following codes:

Platform	License
Data ONTAP storage system	KZZTWOJ
NearStore®	TXKMEAK
Gateway	PLFQNUJ

- You must have an F800 series storage system, F800 series cluster, or better.

Note

FAS200 series storage systems and NearStore storage systems do not support SyncMirror configurations.

- You must have DS14 or FC9 disk shelves.
- You must connect disk shelves to the storage system in a configuration that supports mirrored volumes. For more information about connecting disk shelves to support mirrored aggregates or traditional volumes, see the *Storage Management Guide*, *System Configuration Guide*, *Cluster Guide*, *Fibre Channel DiskShelf14 Hardware Guide*, and *Fibre Channel StorageShelf FC9 Hardware Guide*.

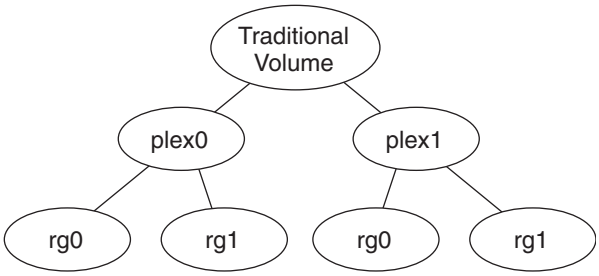
Note

The SyncMirror function is not supported on cluster configurations that use the Servernet cluster card.

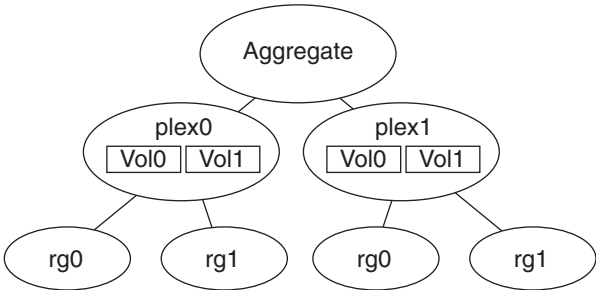
What mirrored aggregates and traditional volumes are

A mirrored aggregate or traditional volume is a single WAFL storage file system with two physically separated and synchronously up-to-date copies on disks. These copies are called plexes. Data ONTAP automatically names the first plex *plex0* and the second plex *plex1*.

Each plex is a physical copy of the same WAFL file system and consists of one or more RAID groups. Because SyncMirror duplicates complete WAFL file systems, you cannot use the SyncMirror feature with a FlexVol volume—only aggregates (including all contained FlexVol volumes) or traditional volumes are supported. The following diagram illustrates the concept of the plexes in a traditional SyncMirror relationship.



The following diagram illustrates the concept of the plexes in an aggregate SyncMirror relationship. Note that each plex includes a copy of the FlexVol volumes contained in the aggregate.



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 separates spare disks from a collection of disks, called a disk pool, into two disk pools, *pool0* and *pool1*. When assigning a disk to a pool, Data ONTAP determines the shelf for the disk and makes sure that the disks in pool0 are from different shelves than the disks in pool1. So before enabling SyncMirror, make sure disks are installed in at least two shelves and the shelves are connected to the storage system with separate cables and adapters. Disk pools must be physically separate to ensure high availability of the mirrored volume.

Disks from pool0 are used to create plex0 while disks from pool1 are used to create plex1.

Plex pool assignment

Plexes local to the host node in a cluster must be connected to the disk pool named pool0. Pool0 consists of the storage attached to host adapters in slots 3 through 7. Pool rules for MetroCluster configurations that use switches are different. See the *System Configuration Guide* for more information about storage system slot assignments.

Enabling and disabling the mirroring license

Enabling the mirroring license

The license code name for the mirroring feature is syncmirror_local. To enable the mirroring license, complete the following steps.

Step	Action
1	Enter the following command: `license add xxxxxxx` *xxxxxxx* is the syncmirror_local license code you purchased.
2	Reboot the system.

Disabling the mirroring license

You cannot disable the mirroring license if mirrored volumes or mirrored aggregates exist and are online. Before disabling the mirroring license, you must take one of the plexes offline for each mirrored volume or mirrored aggregate and destroy it.

Disabling if you have traditional volumes: To disable the mirroring license, complete the following steps.

Step	Action
1	For each mirrored volume, decide which plex you want to take offline. **Note** Every mirrored volume must have one plex taken offline and destroyed before you can disable the mirroring license.

Step	Action
2	Take offline each of the plexes you decided to take offline by entering the following command: **`vol offline plex-name`** *plex-name* is the name of one of the mirrored plexes. **Note** Only one plex at a time can be taken offline.
3	Destroy the plex you took offline by entering the following command: **`vol destroy plex-name`** *plex-name* is the name of one of the mirrored plexes.
4	Enter the following command: **`license delete syncmirror_local`**

Disabling if you have mirrored aggregates: To disable the mirroring license, complete the following steps.

Step	Action
1	For each mirrored aggregate, decide which plex you want to take offline. **Note** Every mirrored aggregate must have one plex taken offline and destroyed before you can disable the mirroring license.
2	Take offline each of the FlexVol volumes in the plexes you decided to take offline by entering the following command: **`vol offline vol-name`** *vol-name* is the name of the FlexVol volume. **Note** Only one FlexVol volume at a time can be taken offline.

Step	Action
3	Remove the FlexVol volumes from the plexes you decided to take offline by entering the following command: **`vol destroy vol-name`** *vol-name* is the name of the FlexVol volume. **Note** Only one FlexVol volume at a time can be taken offline.
4	Take offline each of the plexes you decided to take offline by entering the following command: **`aggr offline plex-name`** *plex-name* is the name of one of the mirrored plexes. **Note** Only one plex at a time can be taken offline.
5	Destroy the plex you took offline by entering the following command: **`aggr destroy plex-name`** *plex-name* is the name of one of the mirrored plexes.
6	Enter the following command: **`license delete syncmirror_local`**

Creating mirrored aggregates and traditional volumes

Two ways to create mirrored aggregates or traditional volumes

You can create a mirrored aggregate or traditional volume in the following ways:

- You can create a new aggregate or traditional volume that has two plexes.
- You can add a plex to an existing unmirrored aggregate or traditional volume.

Note

A mirrored aggregate or traditional volume cannot have more than two plexes.

How Data ONTAP selects disks

Regardless of which way you create a mirrored aggregate or traditional volume, when you create it or add disks to it, Data ONTAP determines what disks to use. Data ONTAP uses the following policies when selecting disks for mirrored aggregate or traditional volumes:

- Disks selected for each plex 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.

Note

When creating an unmirrored volume, Data ONTAP selects disks from the plex which has the most available disks. You can override this selection policy by specifying the disks to use.

Disk selection policies if you select disks

Data ONTAP allows you to select disks when creating mirrored aggregate or traditional volumes or adding disks to them. You need to follow the same disk selection policies that Data ONTAP follows when selecting disks for mirrored aggregate or traditional volumes. See the previous section, “How Data ONTAP selects disks” on page 409.

See "Creating a mirrored aggregate or traditional volume" on page 414 and "Adding a plex" on page 420 for information about creating mirrored aggregate or traditional volumes with disks you select.

Data ONTAP names plexes

Regardless of how a mirrored aggregate or traditional volume is created, Data ONTAP names the plexes of the mirrored aggregate or traditional volume. For more information about the plex naming convention, see "What mirrored aggregates and traditional volumes are" on page 403.

Viewing plex and disk pools

Viewing the disks in a plex or the spare disks in a disk pool is useful for adding disks to a plex or for understanding 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 **`aggr status -r`** or **`vol status -r`**

Example: In this example, the `aggr status -r` command is used to view the disks in plexes and spare disks in disk pools:

```
system1> aggr status -r
Aggregate vol0 (online, raid4) (block checksums)
Plex /vol0/plex0 (online, normal, active, pool1)
RAID group /vol0/plex0/rg0 (normal)

RAID Disk Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks) Ph)
--------- ------  ------------- ---- ---- ---- ----- --------------    ---
parity    9a.16   9a    1   0   FC:A   1  FCAL 10000 34000/69632000    34
data      9a.17   9a    1   1   FC:A   1  FCAL 10000 600/1228800       76
data      9a.20   9a    1   4   FC:A   1  FCAL 10000 34000/69632000    34

Aggregate GreG (online, raid4) (block checksums)
Plex /GreG/plex0 (online, normal, active, pool1)
RAID group /GreG/plex0/rg0 (normal)

RAID Disk Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)    Ph)
--------- ------  ------------- ---- ---- ---- ----- --------------    ---
parity    9a.18   9a    1   2   FC:A   1  FCAL 10000 34000/69632000    34
data      9a.19   9a    1   3   FC:A   1  FCAL 10000 34000/69632000    34


Pool1 spare disks

RAID Disk       Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)    Ph)
---------       ------  ------------- ---- ---- ---- ----- --------------    ---
Spare disks for block or zoned checksum traditional volumes or aggregates
spare           9a.24   9a    1   8   FC:A   1  FCAL 10000 34000/69632000    34
spare           9a.29   9a    1   13  FC:A   1  FCAL 10000 34000/69632000    34

Pool0 spare disks (empty)

Partner disks

RAID Disk       Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)    Ph)
---------       ------  ------------- ---- ---- ---- ----- --------------    ---
partner         9b.25   9b    1   9   FC:B   1  FCAL 10000 0/0               34
partner         9b.16   9b    1   0   FC:B   1  FCAL 10000 0/0               34
partner         9b.17   9b    1   1   FC:B   1  FCAL 10000 0/0               34
partner         9b.21   9b    1   5   FC:B   1  FCAL 10000 0/0               34
partner         9b.18   9b    1   2   FC:B   1  FCAL 10000 0/0               34
partner         9b.22   9b    1   6   FC:B   1  FCAL 10000 0/0               34
tpubs-cf1>
```

Example: In this example, the `vol status -r` command is used to view the disks in plexes and spare disks in disk pools:

```
system1> vol status -r
Volume vol0 (online, raid4) (block checksums)
  Plex /vol0/plex0 (online, normal, active, pool1)
    RAID group /vol0/plex0/rg0 (normal)

      RAID Disk Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)    Ph)
      --------- ------  ------------ ---- ---- ---- ----- --------------    ---
      parity    9a.16   9a    1   0   FC:A   1  FCAL 10000 34000/69632000    34
      data      9a.17   9a    1   1   FC:A   1  FCAL 10000 600/1228800       76
      data      9a.20   9a    1   4   FC:A   1  FCAL 10000 34000/69632000    34


Aggregate aggrz (online, raid4) (block checksums)
  Plex /aggrz/plex0 (online, normal, active, pool1)
    RAID group /aggrz/plex0/rg0 (normal)

      RAID Disk Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)    Ph)
      --------- ------  ------------ ---- ---- ---- ----- --------------    ---
      parity    9a.25   9a    1   9   FC:A   1  FCAL 10000 34000/69632000    34
      data      9a.26   9a    1   10  FC:A   1  FCAL 10000 34000/69632000    34
      data      9a.27   9a    1   11  FC:A   1  FCAL 10000 34000/69632000    34
      data      9a.28   9a    1   12  FC:A   1  FCAL 10000 34000/69632000    34


Pool1 spare disks
RAID Disk       Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)    Ph)
---------       ------  ------------ ---- ---- ---- ----- --------------    ---
Spare disks for block or zoned checksum traditional volumes or aggregates
spare           9a.24   9a    1   8   FC:A   1  FCAL 10000 34000/69632000    34
spare           9a.29   9a    1   13  FC:A   1  FCAL 10000 34000/69632000    34


Pool0 spare disks (empty)

Partner disks
RAID Disk       Device  HA  SHELF BAY CHAN Pool Type  RPM  Used (MB/blks)    Ph)
---------       ------  ------------ ---- ---- ---- ----- --------------    ---
partner         9b.25   9b    1   9   FC:B   1  FCAL 10000 0/0               34
partner         9b.16   9b    1   0   FC:B   1  FCAL 10000 0/0               34
partner         9b.17   9b    1   1   FC:B   1  FCAL 10000 0/0               34
partner         9b.21   9b    1   5   FC:B   1  FCAL 10000 0/0               34
partner         9b.18   9b    1   2   FC:B   1  FCAL 10000 0/0               34
partner         9b.22   9b    1   6   FC:B   1  FCAL 10000 0/0               34
partner         9b.23   9b    1   7   FC:B   1  FCAL 10000 0/0               34
```

For detailed information

The following sections discuss the ways you can create mirrored aggregate or traditional volumes:

- "Creating a mirrored aggregate or traditional volume" on page 414
- "Adding a plex" on page 420

Creating a mirrored aggregate or traditional volume

Methods for creating a mirrored volume

You can create a mirrored aggregate or traditional volume using one of the following methods:

- Let Data ONTAP select the disks used to mirror the aggregate or traditional volume.
- Select the disks used to mirror the aggregate or traditional 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 aggregate or traditional volume by letting Data ONTAP select the disks used, complete the following step.

Step	Action
1	Enter the following command: **{ aggr | vol } create { *aggrname* | *volname}* -m *ndisks*[*@disk-size*]** *aggrname* is the name for the new aggregate. *volname* is the name for the new traditional volume (without the /vol/ prefix). `-m` specifies a mirrored aggregate or traditional volume to Data ONTAP. *ndisks* is the number of disks to use; 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* specifies the disk capacity, in gigabytes. There must be enough disks of this size available to both disk pools 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, the following command creates a mirrored aggregate that has two plexes, each plex containing three disks: `aggr create aggrA -m 6`

You select the disks

If you select the disks used to create a mirrored aggregate or traditional volume, you must ensure that you choose a correct number and size of disks from each pool.

To create a mirrored aggregate using disks you select, complete the following step.

Step	Action
1	Enter the following command: **`aggr create aggrname -m -d disk-list -d disk-list`** *aggrname* is the name of the aggregate you are mirroring. `-m` specifies a mirrored aggregate to Data ONTAP. *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, the following command creates a mirrored aggregate named aggrA with disks 6.1 and 6.2 on one plex and disks 8.1 and 8.2 on the other plex: `aggr create aggrA -m -d 6.1 6.2 -d 8.1 8.2`

To create a mirrored traditional volume using disks you select, complete the following step.

Step	Action
1	Enter the following command: **`vol create volname -m -d disk-list -d disk-list`** *volname* is the name of the volume you are mirroring (without the /vol/ prefix). `-m` specifies a mirrored volume to Data ONTAP. *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, the following command creates a mirrored volume named vol A with disks 6.1 and 6.2 on one plex and 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

You can preview what disks Data ONTAP would select if it were to create a mirrored aggregate or traditional volume.

To create a mirrored aggregate by previewing the disks Data ONTAP would use in the form of a sample command, complete the following steps.

Step	Action	
1	Enter the following command: **`aggr create aggrname -n -m ndisks[@disk-size]`** *aggrname* is the name of the mirrored volume you are creating. `-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 aggregate to Data ONTAP. *ndisks* is the number of disks to use. *ndisks* must be an even number. *disk-size* specifies the disk capacity, in gigabytes. You must have enough available disks of the size you specify. **Result:** Data ONTAP returns an `aggr create` command that specifies the disks.	
2	**If you want to...**	**Then...**
	Use the disks specified by Data ONTAP	At the prompt, enter the `aggr create` command specified.
	Substitute other disks for one or more disks specified by Data ONTAP	At the prompt, enter the `aggr create` command and substitute other disks for one or more of the disks specified.

Example: In this example, the following command previews the command that Data ONTAP would use when creating a 4-disk mirrored aggregate called aggrB:

```
aggr create aggrB -n -m 4
```

Data ONTAP returns the following command:

```
aggr create aggrB -m -d 5.1 5.3 -d 8.3 8.4
```

To create a mirrored traditional volume by previewing the disks Data ONTAP would use in the form of a sample command, complete the following steps.

Step	Action	
1	Enter the following command: **`vol create volname -n -m ndisks[@disk-size]`** *volname* 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. *ndisks* must be an even number. *disk-size* specifies the disk capacity, in gigabytes. You must have enough 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, the following command previews 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

Mirroring an unmirrored aggregate or traditional volume that uses mixed disk capacities

If the unmirrored volume that you want to mirror uses disks of different capacities, Data ONTAP can select 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 409 for the rules Data ONTAP uses to select disks.

Methods for adding a plex to an unmirrored aggregate or traditional volume

You can add a plex to an existing unmirrored aggregate or traditional volume as follows:

- Let Data ONTAP select the disks used to mirror the volume.
- Select the disks used to mirror the aggregate or traditional 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

Letting Data ONTAP select the disks is the easiest method of mirroring, but you do not have control over which disks Data ONTAP selects.

To mirror an existing aggregate or traditional volume by letting Data ONTAP select the disks used to mirror the volume, complete the following step.

Step	Action
1	Enter the following command: **`{ aggr | vol } mirror { aggrname | volname }`** *aggrname* is the name of the aggregate you are mirroring. *volname* is the name of the traditional 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 choose a correct number and size of disks from a different pool than that used by the aggregate or traditional volume plex that you are mirroring.

To mirror an existing aggregate or traditional volume using disks you select, complete the following step.

Step	Action
1	Enter the following command: **`{ aggr | vol } mirror { aggrname | volname } -d disk-list`** *aggrname* is the name of the aggregate you are mirroring. *volname* is the name of the traditional 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

You can preview what disks Data ONTAP would select if it were to mirror the aggregate or traditional volume.

To mirror an existing aggregate or traditional volume by previewing disks Data ONTAP would use in the form of a sample command, complete the following steps.

Step	Action
1	Enter the following command: **`{ aggr | vol } mirror { aggrname | volname } -n`** *aggrname* is the name of the aggregate you are mirroring. *volname* is the name of the traditional volume you are mirroring (without the /vol/ prefix). `-n` specifies to Data ONTAP not to create the mirrored aggregate or traditional volume, but show the disks that it would use if it did create the mirrored aggregate or traditional volume. **Result:** Data ONTAP returns an `aggr mirror` command or a `vol mirror` command that specifies the necessary number and type of disks you can use to mirror the aggregate or traditional volume.

Step	Action	
2	**If you want to...**	**Then...**
	Use the disks specified by Data ONTAP	Enter the `aggr mirror` command or `vol mirror` command specified.
	Substitute other disks for one or more disks specified by Data ONTAP	Enter the `aggr mirror` command or `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 aggrA aggregate:

```
aggr mirror aggrA -n
```

Data ONTAP returns the following command:

```
aggr mirror aggrA -d 8.1 8.2 8.3
```

In this example, use the following command to preview the disks that Data ONTAP would use when mirroring the volA traditional 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 aggregate or traditional volumes

Rules to follow when adding disks

When you add disks to a mirrored aggregate or traditional volume, you must follow these rules:

- The number of disks must be even, and the disks must be equally divided between the two plexes.
- The disks for each plex must come from different disk pools.
- The disks you add must have equivalent bytes per sector (bps) sizes.
- The disks must have capacity equal to or greater than existing disks in the plexes.

Note

If you add a larger capacity disk, Data ONTAP downsizes the larger disk to the size of the disks in the volume.

Methods for adding disks

You can add disks to mirrored aggregate or traditional 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	Enter the following command: **{ aggr | vol } add { *aggrname* | *volname* } *ndisks*[@*disk-size*]** *aggrname* is the name of the aggregate to which you are adding disks. *volname* is the name of the traditional 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* specifies the disk capacity, in gigabytes. Half these disks must be in each disk pool. **Note** If you use the `disk-size` option, first determine the size of spare disks in the disk pools using the `aggr status -r` or `vol status -r` command.

You select the disks

If you select the new disks to add to the mirrored aggregate or traditional volume, you must ensure that you add the correct number of disks of the correct sizes. The disks must have the same checksum compatibility, and disks for each plex must be in different pools.

To add new disks to a mirrored aggregate or traditional volume using disks you select, complete the following steps.

Step	Action
1	Enter the following command to list available spare disks: **{ aggr | vol } status -r**
2	Using the rules listed in “Rules to follow when adding disks” on page 423, choose the spare disks you want to add.

Step	Action

3	Enter the following command:
	{ aggr | vol } add { *aggrname* | *volname* } -d *disk-list* -d *disk-list*
	aggrname is the name of the aggregate to which you are adding disks. *volname* is the name of the traditional 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 aggrD mirrored aggregate:
	`aggr add aggrD -d 6.1 -d 8.1`
	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 traditional volume:
	`vol add volD -d 6.1 -d 8.1`

You select the disks with assistance from Data ONTAP

You can preview what disks Data ONTAP would select if it were to add new disks to the aggregate or traditional 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">Enter the following command:
{ aggr | vol } add { aggrname | volname } -n ndisks[@disk-size]
aggrname is the name of the mirrored aggregate to which you are adding disks. volname is the name of the mirrored traditional 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 specifies the disk capacity, in gigabytes. Half these disks must be in each disk pool.
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 an <code>aggr add</code> or <code>vol add</code> command that specifies the disks it would add to each plex of the mirrored volume.</td></tr>
<tr><td rowspan="3">2</td><td>If you want to...</td><td>Then...</td></tr>
<tr><td>Use the disks specified by Data ONTAP</td><td>Enter the <code>aggr add</code> or <code>vol add</code> command specified.</td></tr>
<tr><td>Substitute other disks for one or more disks specified by Data ONTAP</td><td>Enter the <code>aggr add</code> or <code>vol add</code> command and substitute other disks for one or more of the disks specified.</td></tr>
</table>

Example: In this example, the following command previews the disks that Data ONTAP would use when adding two new 36-GB disks to the aggrA mirrored aggregate:

```
aggr add aggrA -n 2@36
```

Data ONTAP returns the following command:

```
aggr add aggrA -d 6.4 -d 8.6
```

Example: In this example, the following command previews 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 an aggregate or traditional volume that is being created and Data ONTAP needs to zero out one or more of the disks targeted to the aggregate or traditional 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 aggregate or traditional volume has failed.
  - Resyncing—The plex contents are being resynchronized with the contents of the other plex of the aggregate or traditional 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 **`aggr status -r`** or **`vol status -r`**

Changing the plex state

Data ONTAP specifies the state of a plex as resyncing when synchronizing the two plexes of a mirrored aggregate or traditional volume. For example, when you create a mirrored aggregate or traditional volume by adding a plex to an existing unmirrored aggregate or traditional 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: **{ aggr | vol } online | offline *plexname***

Splitting mirrored aggregates or traditional volumes

What splitting a mirrored aggregate or traditional volume is

Splitting a mirrored aggregate or traditional volume removes the relationship between its two plexes and creates two independent unmirrored aggregates or traditional volumes. After splitting, the new aggregate or traditional volume is in a restricted state and the original aggregate or traditional volume stays in an online state.

Note

To mirror one or both of these aggregates or traditional volumes, see "Adding a plex" on page 420. To rejoin the aggregates or traditional volumes that you split, see "Rejoining split aggregates or traditional volumes" on page 432.

Why you might split a mirrored aggregate or traditional volume

You might split a mirrored aggregate or traditional volume for one of the following reasons:

- You want to stop mirroring an aggregate or traditional volume. This is an alternative to the procedure described in "Removing and destroying a plex" on page 434.
- You want to move a mirrored aggregate or traditional volume to another location.
- You want to modify the mirrored aggregate or traditional volume, and test the modification before applying it. You can apply and test the modifications and test on the split-off copy of the plex, then apply those changes to the untouched original plex.

Note

You do not need to stop applications that are running before splitting a mirrored aggregate or traditional volume.

Splitting a mirrored aggregate or traditional volume

To split a mirrored aggregate or traditional volume into two independent unmirrored aggregates or traditional volumes, complete the following step.

Note

Ensure that both plexes of the mirrored aggregate or traditional volume you are splitting are online and operational. For information about viewing the state of a plex, see "Changing the state of a plex" on page 428.

Step	Action

1	Enter the following command:
	`{ aggr | vol } split plexname { aggrname | volname }`
	plexname is the name of one of the plexes in the mirrored volume.
	aggrname is the name of the new aggregate. *volname* is the name of the new traditional volume.
	Example: In this example, use the following command to split plex0 from mirrored aggregate aggr0 and name the new aggregate aggrNew:
	`aggr split aggr0/plex0 aggrNew`
	After splitting, there are two unmirrored aggregates, aggr0 and aggrNew.
	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 aggregate or traditional volume has two plexes, plex0 and plex1. After splitting, the new unmirrored aggregate or traditional volume with the new name has one plex, plex0. The new unmirrored aggregate or traditional volume with the original name also has one plex, either plex0 or plex1.

The plex name for an unmirrored aggregate or traditional volume is unimportant because the aggregate or traditional volume has only one plex. If you remirror one of these unmirrored aggregates or volumes, the resulting plex names will always be plex0 and plex1.

Rejoining split aggregates or traditional volumes

Why you would rejoin split aggregates or traditional volumes

You might rejoin split aggregates or traditional volumes because you were forced to split them. This would be the case if you have configured a cluster in a MetroCluster configuration and a disaster breaks the cluster relationship.

Rejoining a split aggregate or traditional volume

When you rejoin a split traditional volume, Data ONTAP mirrors the data from one volume to the other and destroys data that existed on that volume before the rejoin. Likewise, when you rejoin a split aggregate, Data ONTAP mirrors the data from one aggregate to the other and destroys data that existed on that aggregate before the rejoining.

To rejoin a split aggregate or traditional volume, complete the following steps.

Step	Action
1	Determine the aggregate or traditional volume whose data you will keep and the aggregate or traditional volume whose data will be overwritten.
2	If the volume or aggregate whose data will be overwritten is online, take it offline by entering the following command for each volume or aggregate that was split: **`{aggr | vol} offline {`*`aggrname`*` | `*`volname`*`}`** *aggrname* is the name of the aggregate. *volname* is the name of the traditional volume. **Note** An error message appears if the aggregate or volume is already offline.

Step	Action
3	Re-create the mirrored aggregate or traditional volume by entering the following command for each aggregate or traditional volume that was split: **{aggr | vol} mirror {*aggrname1* |*volname1*} -v {*aggrname2* |*volname2*}** *aggrname1* is the name of the aggregate whose data will be kept. *volname1* is the name of the traditional volume whose data will be kept. *aggrname2* is the name of the aggregate whose data will be overwritten by *aggrname1*. *volname2* is the name of the traditional volume whose data will be overwritten by *volname2*.

Removing and destroying a plex

Why you might remove and destroy a plex

You might remove a plex from a mirrored aggregate or traditional 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 aggregate or traditional 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.

You might also remove a plex from a mirrored aggregate or traditional volume and destroy it if you want to stop mirroring the aggregate or traditional volume.

Removing and destroying a plex

You can remove and destroy a plex from a mirrored aggregate or from a traditional volume.

Destroying a plex from a traditional volume: To remove and destroy a plex from a traditional 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
```

Destroying a plex from a mirrored aggregate: Removing and destroying a plex from a mirrored aggregate requires the additional steps of taking FlexVol volumes in the plex offline and destroying them. To remove and destroy a plex from a mirrored aggregate, complete the following steps.

Step	Action
1	Take offline all of the FlexVol volumes in the plexes you decided to take offline by entering the following command: **`vol offline vol-name`** *vol-name* is the name of the FlexVol volume. **Note** Only one FlexVol volume at a time can be taken offline.
2	Remove the FlexVol volumes from the plexes you decided to take offline by entering the following command: **`vol destroy vol-name`** *vol-name* is the name of the FlexVol volume. **Note** Only one FlexVol volume at a time can be taken offline.
3	Take offline all of the plexes you decided to take offline by entering the following command: **`aggr offline plex-name`** *plex-name* is the name of one of the mirrored plexes. **Note** Only one plex at a time can be taken offline.
4	Destroy the plex you took offline by entering the following command: **`aggr destroy plex-name`** *plex-name* is the name of one of the mirrored plexes.

Result of removing and destroying a plex

Removing and destroying a plex from a mirrored aggregate or traditional volume results in an unmirrored aggregate or traditional volume, because the aggregate or traditional 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 aggregate or traditional volumes

When you should compare plexes

Because the plexes of mirrored aggregate and traditional volumes are almost always synchronized, you rarely need to compare the plexes of a mirrored aggregate or traditional volume. If you do need to compare the plexes of a mirrored aggregate or traditional volume, Data ONTAP enables you to do the following tasks:

- Compare plexes

  Note
  Comparing plexes might affect storage system performance.

- Stop comparing plexes
- Suspend and resume comparing plexes
- View the progress of the comparison

Comparing plexes

The mirrored aggregate or traditional volume must be online before you can compare the plexes. When comparing the two plexes of a mirrored aggregate or traditional volume, you can choose one of the following options:

- Data ONTAP compares plexes and corrects differences it finds.
- Data ONTAP compares plexes without correcting differences. If you choose this option, you must determine which plex is the good one and synchronize the other plex to it. This process can be difficult and can use advanced Data ONTAP commands. Consult Technical Support before correcting differences using this option.

To compare the two plexes of a mirrored aggregate or traditional volume, complete the following step.

Step	Action	
1	**If you...**	**Then...**
	Want Data ONTAP to correct differences	Enter the following command: **{ aggr | vol } verify start { *aggrname* | *volname* }** *aggrname* is the name of the mirrored aggregate whose plexes you are comparing. *volname* is the name of the mirrored volume whose plexes you are comparing. If neither *aggrname* nor *volname* is specified, the plexes of all online mirrored volumes are compared.
	Do not want Data ONTAP to correct differences	Enter the following command: **{ aggr | vol } verify start { *aggrname* | *volname* } -n** If neither *aggrname* nor *volname* is specified, Data ONTAP compares the plexes of a mirrored aggregate or traditional volume for all aggregates or traditional volumes.

Stopping plex comparison

You might need to stop Data ONTAP from comparing mirrored aggregate or traditional volume plexes. For example, you might stop verifying because it affects storage system performance. To stop Data ONTAP from comparing plexes, complete the following step.

Step	Action
1	Enter the following command: **{ aggr | vol } verify stop { *aggrname* | *volname* }** If *aggrname* or *volname* is not specified, Data ONTAP stops comparing plexes for all aggregates or traditional volumes.

Suspending and resuming plex comparison

Rather than stopping Data ONTAP from comparing the plexes of a mirrored aggregate or traditional volume, you can suspend the comparison of plexes. The comparison remains suspended until you resume it, stop it, or reboot the system.

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: **{ aggr | vol } verify suspend { *aggrname* | *volname* }** If *aggrname* or *volname* is not specified, Data ONTAP suspends the comparison of plexes for all aggregates or volumes.
	Resume the comparison	Enter the following command: **{ aggr | vol } vol verify resume { *aggrname* | *volname* }** If *aggrname* or *volname* is not specified, Data ONTAP resumes the comparison of plexes for all aggregates or volumes.

Viewing the status of a plex comparison

Plex comparison status tells you what percentage of the plex comparison has been completed and whether plex comparison of a mirrored aggregate or traditional volume is suspended.

To view the status of a plex comparison, complete the following step.

Step	Action
1	Enter the following command: **`{ aggr | vol } verify status { aggrname | volname }`** If *aggrname* or *volname* is not specified, Data ONTAP shows the status of all aggregates or volumes whose plexes are currently being compared.

Database Protection Using NVFAIL 9

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 442
- "Enabling and disabling database file protection" on page 444
- "Using the nvfail_rename file for additional database protection" on page 445

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 boot time, 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 system 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 system 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.

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.

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 storage system'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. You can list any number of files. **Example:** `/vol/vol1/home/dbs/oracle-WG73.dbf`
3	Save the file.

Virus Protection for CIFS 10

About this chapter

This chapter discusses the virus scan feature for CIFS and how to configure it to scan files on a Data ONTAP storage system.

Topics in this chapter

This chapter discusses the following topics:

- "Understanding CIFS virus protection" on page 448
- "Setting up and starting virus scanning" on page 450
- "Specifying file types to be scanned" on page 453
- "Excluding file types to be scanned" on page 455
- "Specifying shares for scanning" on page 459
- "Displaying the scanner list" on page 465
- "Checking vscan information" on page 466
- "Setting and resetting the virus scan request timeout" on page 467
- "Allowing file access when the scan cannot be performed" on page 468
- "Controlling vFiler usage of host storage system virus scanners" on page 469
- "Checking the status of virus-scanning options" on page 471
- "Stopping a virus scanner session" on page 472
- "Resetting the scanned files cache" on page 473
- "Enabling virus scan messages to CIFS clients" on page 474

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 compliant antivirus applications to provide on-access virus scanning of files on a storage system. 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 Data ONTAP-compliant antivirus application of your choice. When you enable the virus-scanning process through Data ONTAP on the storage system, the virus-scanning application tells the system to send file scanning requests.

The virus-scanning application watches for requests from the storage system. Whenever the types of files you specify are opened or changed on the storage system, Data ONTAP sends the PC client a request to scan the file.

The Data ONTAP virus-scanning process can scan multiple storage systems 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 systems, contact the manufacturer of your virus-scanning application.

Obtaining compliant anti-virus software

In order to successfully scan for viruses on the storage system, the anti-virus application installed on your PC client must be a version customized for Data ONTAP storage systems and supplied by a supported anti-virus software partner.

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. This 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 453.

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 storage systems.

During installation of the commercial virus scan software on the selected PCs, you indicate the Data ONTAP storage systems 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 a Data ONTAP-compliant virus scan software.
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 systems on which they will conduct virus scanning.
3	Install the Data ONTAP-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 system 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 storage system.

Step	Action
5	Leave the virus-scanning client on and connected to the storage system(s) 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 storage system, 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 system. In the system 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 storage system, 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 belongs to 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 storage system.

Displaying files to be scanned

To display the list of extensions of file types to be scanned on the storage system, 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 449.

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 storage system, prefer to specify what file types to exclude, rather than include, in a virus scan. This practice allows you to pass file types assumed to be safe, at the same time ensuring that any new, unfamiliar file type stored on the system does get 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 storage system.

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 system 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 system. 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 storage system, 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 460
- "Turning scanning on or off for read-only access for shares" on page 462
- "Adding shares with virus scanning turned on or off" on page 463

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 it is used only by trusted users, the files are restricted to read-only mode, or speed of access is more important than safety. Virus scanning for a share is turned on by default.

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 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 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. This increases the speed of accessing the files. A share has virus scanning turned on by default.

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 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 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. A share has virus scanning turned on by default.

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 Access Control List (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 system. To display the scanner list, complete the following step.

Step	Action
1	Enter the following command: **vscan scanners** **Result:** The system 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 system.

The *Reqs* field displays the total number of requests sent by the system 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 whether vscan is on, 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 storage system.

To check the status of the vscan appliance, complete the following step.

Step	Action
1	Enter the following command: **vscan** **Result:** The system 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.

```
systemA> 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 represents this connection session. The second set of numbers at the end of the output 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 10 seconds.

Note

Prior to Data ONTAP 6.4.3, the default timeout value was 30 seconds. Storage systems that you upgraded from pre-Data ONTAP 6.4.3 releases will still have the 30 second timeout after the upgrade.

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. The recommended setting is between 8 and 12 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 storage system 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 storage system so that it appears as multiple storage systems on the network. Each "storage system" created as a result of the partitioning is called a vFiler, which represents a virtual storage system. A vFiler, using the resources assigned, delivers file services to its clients as a storage system does. For more information about vFilers, see the *MultiStore Administration Guide.*

About controlling virus scanner usage

By default, vFilers can scan files using the virus scanners that are connected to the host storage system. You can register a virus scanner in one of two ways:

- The virus scanner registers with each vFiler individually. In this way, each vFiler authenticates the virus scanner and the vFiler appears as a separate machine to the virus scanner.
- The virus scanner registers with only the physical storage system. In this way, only the storage system authenticates the virus scanner.

Note

The virus scanner accesses all files through the storage system, not through each vFiler.

If you have several vFilers, each owned by a separate organization, you might want to have the virus scanner register with each vFiler if you have security concerns.

You can control whether vFilers are allowed to use the host storage system'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 storage system, 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 storage system 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 storage system 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 storage system, 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 storage system 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.

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 Snapshot copies.

aggregate A manageable unit of RAID-protected storage, consisting of one or two plexes, that can contain one traditional volume or multiple FlexVol volumes.

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 storage system's serial port and is used to monitor and manage storage system 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 storage system and in a UNIX file system.

local tape device A program-based functionality associated with a tape drive that is directly attached to a storage system 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 storage system.

NDMP Network Data Management Protocol. A protocol that allows Data ONTAP storage systems 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 storage system, used for logging incoming write data and NFS requests. Improves system performance and prevents loss of data in case of a storage system or power failure.

Open Systems platform A 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 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 User ID (UID) or group ID (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. Storage systems running Data ONTAP software 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 storage system that is performing a tape operation.

root volume The volume that contains information that controls the entire storage system, usually in the /etc/rc file.

secondary storage system A storage system to which data is backed up by SnapVault.

share A directory or directory structure on the storage system 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 storage system.

SnapRestore Software that restores an entire volume to the state recorded in a previously taken Snapshot copy.

Snapshot copy An online, read-only copy of the entire file system that protects against accidental deletions or modifications of files without duplicating file contents. Snapshot copies enable users to restore files and enable administrators to back up the system to tape while it is in use.

Snapshot reserve The portion of a volume's disk space that is reserved for Snapshot copies.

source storage system The storage system 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.

volume A file system.

volume copy A way of copying both data in the active file system and data in Snapshot copies from one volume to another.

Index

Symbols

/etc/hosts.equiv file 384
/etc/log/snapmirror 154
/etc/nvfail_rename file 442
/etc/snapmirror.allow file
    purpose 113
    sample 114
/etc/snapmirror.conf file
    arguments field defaults 124
    dest_filer field 119
    format 118
    purpose 116
    sample 123
    SnapMirror scheduling in 138

A

access to remote data 83
aggregates
    creating SnapLock 366
    SnapLock and 366
aggregates, mirrored understanding 402
asynchronous mirroring 79
Autosupport and SnapLock 365

B

backups
    incremental, after SnapRestore use 75
    online
        advantages 8
        disadvantages 8
        methods 8
    SnapLock compliance volumes 213
    SnapLock enterprise volumes 215
    to tape, advantages of 8
    using Open Systems SnapVault for 252
    using SnapMirror for 83
    why necessary 2
bootup, with nvfail enabled 442

C

cascading
    from synchronous SnapMirror 192
    SnapMirror destinations 192
cascading SnapMirror destinations 190
cifs shares command 460, 462, 463
CIFS, virus protection for 447
cifs.snapshot_file_folding.enable option 48
clients
    accessing snapshots from 20
    disabling access to snapshots for 22
    snapshot directory invisible from 22
command conventions xiii
commands
    See also SnapMirror commands
    See also SnapRestore commands
    See also snapshot commands
    See also SnapVault commands
    See also SyncMirror commands
    See also volume copy commands
    cpio 378
    df (finds snapshot disk consumption) 39
    hostname 119
    ls -a (shows snapshot files) 21
    ls -lu (shows when snapshots created) 35
    qtree (displays status of qtrees) 136
    quotas on (enables quotas) 169
    rsh (with vol copy abort) 399
    rsh (with vol copy status) 394
    vol copy throttle 398
    vol create -m (creates a pair of mirrored volumes) 415
    vol destroy command (removes a plex from a mirrored pair) 434, 435
    vol split command (splits a mirrored volume) 431
    vol verify start command (compares data in mirrored volumes) 437
    vol verify status command (displays percentage of comparison of data in mirrored volumes completed) 440
    vol verify stop command (stops comparing

data in mirrored volumes) 439
vol verify suspend command (suspends comparison of data in mirrored volumes) 439
compliance clock
about 368
initializing 368
viewing 369
compliance volume, SnapLock 211
compliance, SEC240.17a-4 211
configuration files
/etc/nvfail_rename 442
/etc/snapmirror.conf 116
avoiding reversion of 64
conventions in this manual xiii, xiv
copying volumes. *See* volume copy
cpio command, disadvantages of 378

D

data
access to remote, using SnapMirror 83
online recovery 7
protection introduction. *See also* disaster recovery 2
recovery using SnapRestore 61
data migration using SnapMirror 183
database file protection
nvfail warnings 442
renaming files 442
databases, testing using SnapMirror 179
Depending 2
destination volumes
changing names of 209
maximum allowed by SnapMirror 86
df command, displaying free space on disk 39
directories of snapshots
on CIFS clients 21
on NFS clients 20
disaster recovery
using SnapMirror 82, 180
disk space
check snapshot consumption 39
freeing by deleting snapshots 56
how calculated by snap list 41
how snapshots consume 36
percentages used, how calculated 41
reclaiming from snapshots 46
reserving for snapshots 44
saving with file folding 48
dump command, effect on SnapMirror update schedule 208

E

enterprise volume, SnapLock 211
error messages
from volume copying 392
Stale File Handle 67
examples
SnapMirror log files 156
SnapMirror, use of snapshots 142
snapmirror.conf file 123
vol copy throttle command (controls volume copy speed) 398
volume copy start command 391

F

failover mode, of multiple paths 126
Fibre Channel connection, troubleshooting reset messages 111
file access times of snapshots
defined 33
from CIFS clients 35
from NFS clients 35
file extensions
adding to vscan list 453
excluding from a scan 455, 457
for vscan, excluding 455
for vscan, viewing 453
removing from a vscan exclude list 457
removing from vscan 454, 460
replacing extensions to exclude from a scan 456
replacing in vscan list 454
resetting to vscan default exclude list 457
resetting to vscan default list 454
viewing extensions to exclude from a scan 456
file folding
defined 48

disabling 48
starting 48
file versions of snapshots
defined 33
finding all 33
FilerView xiii
files
nvfail_rename 445
protecting with nvfail option 442
restoring from snapshots 24
reverting from snapshots with SnapRestore 71
firewalls, SnapMirror recommendation 102
FlexVol volumes
SnapLock and 366

H

hostname command 119
hourly snapshots 26
HP-UX
See also Open Systems platform
backing up with SnapVault 222

I

initializing SnapMirror 130

L

license add command 245
license add for SnapVault 232, 303
license add for synchronous SnapMirror 95, 402
load balancing, using SnapMirror for 83
log file of SnapMirror transfers 154
LUN (logical unit number), data protection of volumes containing LUNs 15

M

migration of data using SnapMirror 183
mirrored volumes and aggregates
adding disks to 423
adding synchronous mirroring to an existing volume 420
creating at the same time 414
defined 403
removing and destroying a plex 434
removing mirroring between 430
verifying data 437
mirroring
asynchronous 79
synchronous 79
multihop. *See* cascading
multiple paths
failover mode 126
implementing 126
multiplexing mode 126

N

ndmpd on option (turns NDMP service on) 256
newest common snapshot (NCS) 101, 174
NFS clients
accessing snapshots from 20
determining snapshot file access time from 35
nightly snapshots 25
nvfail option
bootup process 442
renaming files for database protection 445
sending warnings for database protection 442
using nvfail_rename file 445
what it does 442
nvfail_rename file 445

O

online data recovery 7
Open Systems
license add command 245
managing 244
restoring data 249
restoring primary storage system 251
secondary storage configuration 241
snapvault snap -x command 246
snapvault start command 245
svrestore command 249
Open Systems platform
definition 220
non-filer systems supported for SnapVault backup 222
Open Systems SnapVault agent, definition 220
operation numbers

using with vol copy abort 399
using with vol.copy.throttle 398
options
cifs.snapshot_file_filding.enable 48
license add 232, 303
snapmirror.checkip.enable (lets SnapMirror resolve host names to IP addresses) 115
snapmirror.enable off (disables SnapMirror) 166
snapmirror.enable on (enables SnapMirror) 128
snapmirror.log.enable 154
snapvault.access 232, 256, 304
snapvault.enable off 343
snapvault.enable on (turns SnapVault on) 232, 233, 241, 255, 304, 347, 358
vol.copy.throttle (controls volume copy speed) 397

P

permissions, for snapshots 20
Plexes
states 428
plexes
adding 420
changing state of 429
comparing 437
removing and destroying 434
stop comparison of 439
suspension and resumption of comparison 439
verifying data 437
viewing status of 428
viewing status of comparison 440
primary storage system
defined 220
filer and Open Systems platforms 222
restoring 251
primary system qtree, definition 220
private networks, for volume copy 380

Q

qtree command
displaying status of qtrees 136
qtrees
displaying status 136
quotas in SnapMirror 130
SnapMirror replication, characteristics of 85
SnapMirror replication, components of 79
SnapMirror replication, initialization guidelines 130
quiesce. *See* snapmirror quiesce command
quotas
enabling on former mirrors 169
quota on command (enables quotas) 169

R

RAID command, vol volume (changes RAID group size) 407
RAID group size, changing (vol volume) 407
recovery from disaster, using SnapMirror 82
remote access to data, using SnapMirror for 83
Remote Shell
using with vol copy 387
using with vol copy status 394
renaming snapshots 59
replicating volumes 77, 82
reset messages, troubleshooting 111
restarting transfers, in SnapMirror 124
restrictions on SnapLock volumes 212
resynchronizing SnapMirror source and mirror 176
retrying transfers, in SnapMirror 124
reverting volumes with SnapRestore 67
root volumes, reverting with SnapRestore 65
rsh command
using with vol copy abort 399
using with vol copy status 394

S

SAN (data protection in the SAN environment) 15
scanning. *See* virus scanning
schedules
for default snapshots 25
for user-defined snapshots 29
strategies for snapshots 28
where to specify for SnapMirror 116
SEC 17a-4 compliance 211
secondary storage system

definition 220
role in a basic SnapVault deployment 223
secondary system qtree, definition 220
shares
adding with virus scanning turned off 463
enabling or disabling vscan 460
enabling or disabling vscan for read-only shares 462
simultaneous 86
simultaneous transfers 86
size, volume, verifying for SnapMirror 382
snap list command (determining which snapshots to delete) 56
snap list -o command (lists origins of qtree snapshots) 281
snap list -q command (lists qtree snapshots) 278
snap restore -t file command (reverts file from snapshot) 71
snap restore -t vol command (reverts volume from snapshot) 68
SnapLock
about 211, 364
aggregates and 366
Autosupport and 365
compliance clock
about 368
initializing 368
viewing 369
compliance volume 211
creating aggregates 366
creating compliance SnapMirror backup 213
creating enterprise SnapMirror backup 215
creating traditional volumes 366
data, moving to WORM state 375
destroying aggregates 374
destroying volumes 373
enterprise volume 211
files, determining if in WORM state 376
FlexVol volumes and 366
how it works 364
licensing 365
replication and 365
restrictions 212
retention dates
extending 375
setting 375
retention periods
default 370
maximum 370
minimum 370
setting 371
when to set 370
SEC 17a-4 compliance 211
types of volumes 211
volume retention periods *See* SnapLock retention periods
when you can destroy aggregates 373
when you can destroy volumes 373
WORM requirements 368
write_verify option 367
SnapLock Compliance, about 364
SnapLock Enterprise, about 364
SnapMirror
/etc/snapmirror.conf file
arguments field defaults 124
dest_filer field (defines destination filer) 119
backing up to tape with 196
blocking transfers 160
cascading destinations
defined 190
from a synchronous SnapMirror 192
restructuring a cascade 193
cascading mirrors 191
checking status (snapmirror status) 144, 153
complete transfer, when to use 130
configuring for Fibre Channel 109
converting mirrors to regular volumes (snapmirror break) 168
copying mirrors from mirrors 191
creating SnapLock compliance backups 213
creating SnapLock enterprise backups 215
enabling quotas on former mirrors (quota on) 169
failover mode 126
Fibre Channel connections 109
Fibre Channel connections, troubleshooting 111
halting a transfer 159
host names, resolving to IP addresses 115

implementing multiple paths 126
license requirement for 77
log file 154
logging transfer history 154
maximum number of destination volumes allowed by 86
migrating data between volumes 183
mirroring volumes 82, 128
multiplexing mode 126
names of snapshots created 141, 142
overview of 82
preserving quotas after converting from a mirror 168
qtree replication, characteristics of 85
qtree replication, components of 79
qtree replication, difference between volume replication and 84
qtree replication, initialization guidelines 130
qtree replication, quotas 130
releasing mirrors 171
replicating synchronous 93
replicating volumes 128
replicating volumes to tape (snapmirror store) 196
requirements 100
restarting replication of volumes to new tape (snapmirror store) 200, 205
restarting transfers, defined 124
restoring volumes from tape (snapmirror retrieve) 201, 205, 206
restricting destination volumes 131
restrictions 100
restrictions on SnapLock destination volume 212
resuming transfers after quiescing 162
retrying transfers, defined 124
schedules, where specified 138
snapmirror.log.enable option 154
specifying destinations
using options snapmirror.access 113
using snapmirror.allow file 114
status messages 144, 153
stopping replication of a volume (snapmirror abort) 159
storage system upgrade order 100
synchronous 93
synchronous running asynchronously 93
synchronous, cascading from 192
synchronous, configuration for optimal performance 94
synchronous, configuring for IP over Fibre Channel 109
synchronous, considerations 94
synchronous, converting from asynchronous 109
synchronous, filers not supported 94
synchronous, optimized connection 94
synchronous, using multiple paths 126
tips for efficient scheduling 128
transfers through firewalls 102
troubleshooting the Fibre Channel connection 111
turning off scheduled updates for filer (snapmirror off) 166
using for database testing 179
using for disaster recovery 180
using multiple paths 126
using with dump command 208
using with snapshots, example 142
verifying volume size 382
snapmirror abort command (halts specified transfer) 159
snapmirror break command (converts mirror to regular volume) 169, 181, 207
SnapMirror commands
options snapmirror.enable off (disables SnapMirror) 166
options snapmirror.enable on (enables SnapMirror) 128
snapmirror abort (halts specified transfer) 159
snapmirror abort (stops replication of a volume) 159
snapmirror break (converts mirror to regular volume) 169, 181, 207
snapmirror destinations (lists required snapshots) 192
snapmirror initialize (performs base-level transfer to mirror) 132
snapmirror off (disables SnapMirror) 166
snapmirror off (turns off scheduled updates for

filer) 166
snapmirror on (enables SnapMirror) 128
snapmirror quiesce (blocks transfers) 160
snapmirror release (releases a volume from mirror status) 171
snapmirror resume (resumes transfers after quiescing) 162
snapmirror resync (resynchronizes a source to a mirror) 176
snapmirror retrieve (restores volumes from tape) 201, 205, 206
snapmirror status (checks data replication status) 153
snapmirror status (checks data transfer status) 144
snapmirror store (replicates volume to tape) 196
snapmirror update (performs unscheduled incremental update of mirror) 138
snapmirror use (restarts replication to new tape) 200, 205
snapmirror destinations command (lists required snapshots) 192
snapmirror initialize command (performs base-level transfer to mirror) 132
snapmirror quiesce command (blocks transfers) 160
snapmirror release command (releases a volume from mirror status) 171
snapmirror resume command (resumes transfers after quiescing) 162
snapmirror resync command (resynchronizes a source to a mirror) 176
snapmirror retrieve command (restores volumes from tape) 201, 206
snapmirror store command (replicates volume to tape) 196
snapmirror update command (performs unscheduled incremental update of mirror) 138
snapmirror use command (restarts replication of volumes to new tape) 200, 205
snapmirror.access option (specifies SnapMirror destinations from SnapMirror source) 113
snapmirror.allow file (specifies SnapMirror destinations from SnapMirror source) 114
snapmirror.checkip.enable option (lets SnapMirror resolve host names to IP addresses) 115
snapmirror.conf file. *See* /etc/snapmirror.conf file
snapmirror.log.enable option 154
snapmirror.window_size option 103
SnapRestore
avoiding reversion of configuration files 64
effect on snapshots used 64
how it works 62
license requirement for 61
prerequisites for 63
preserving configuration files using 64
reverting root volumes using 65
reverting volumes using 67
typical applications using 62
using with NFS 67
what it does not revert 62
when to use 63
with previous releases 63
SnapRestore commands
snap restore -t file (reverts file from snapshot) 71
snap restore -t vol (reverts volume from snapshot) 68
snapshot basename, definition 220
snapshot commands
snap create (creates snapshots manually) 32
snap delete (deletes snapshots) 57
snap list (shows disk space used by snapshots) 40, 43
snap list -o (lists origins of qtree snapshots) 281
snap list -q (lists qtree snapshots) 278
snap rename (renames snapshots) 59
snap reserve (sets percent of disk space for) 47
snap sched 26, 29
vol options nosnapdir (makes directories visible or invisible) 23
snapshots
accessing
from CIFS clients 22
from NFS clients 21
automatic types of 25
changing schedules (snap sched) 29
creating manually (snap create) 32

default schedule 25
defined 18
deleting (snap delete) 57
deleting manually 57
deleting when locked 57
determining which to delete 42
df command (displays disk usage) 39
directories
making visible or invisible 23
on CIFS clients 21
on NFS clients 21
directory structure on NFS clients 20
disk consumption by 33, 56
displaying schedules (snap sched) 26
file access times
defined 33
from CIFS clients 35
from NFS clients 35
file names on NFS clients 21
file versions, defined 33
finding all versions of a file 33
hourly 26
how snap list results are calculated 40
invisible directories 21
ls command (shows files in a directory) 21
making directories visible or invisible (vol options nosnapdir) 23
maximum number of 18
names created by SnapMirror 141, 142
newest common 101
nightly 25
permissions to read 20
reclaiming disk space from 46
recovering from accidental deletion 209
renaming (snap rename) 59
reserving space for 40, 44
restoring files from 24
saving disk space with file folding 48
schedules, sample file 29
setting space reserved for 47
showing disk space used (snap list command) 40
showing SnapMirror destinations 192
snap list (shows disk space used by) 40
strategies for scheduling 28
time created 35
used by SnapMirror, example 142
uses of 19
using with SnapRestore 62
weekly 25
SnapVault
advantages 221
backup for qtrees, ending 295
backup relationship, starting a 258
basic configuration for filers 231
primary storage 232, 303
restoring 236
scheduling 235, 306
secondary storage 233, 304, 357
basic deployment 222
changing configuration 246, 283
checking status (snapmirror status) 272
estimating time for initial baseline backup 230
license add command 232, 245, 303
maximum number of snapshots per volume 228
ndmpd on option (turns NDMP service on) 256
non-qtree data, starting a backup of 261
objects backed up on filer systems 221
Open Systems configuration 239
configuring the secondary storage system 241
managing 244
restoring data 249
Open Systems platform objects backed up 221
overview of filer backup process 225
overview of Open Systems backup process 227
planning 228
planning backup schedule and snapshot retention 228
planning primary qtrees or directories and secondary qtrees 228
planning to store more than 251 snapshots per volume 230
primary configuring 246
primary to secondary to tape deployment 223
primary, configuring 247, 267
qtree data, starting a backup of 259
qtree snapshots, listing 278
qtree snapshots, listing origins of 281

restoring data from secondary to primary 289
secondary to SnapMirror deployment 224
secondary, configuring 268
snapshot schedule, unconfiguring 296
snapshots
displaying information about 272
releasing 290, 297, 359
snapvault snap -x command 246
snapvault start command 245
snapvault.access option (controls access) 232, 256, 304
snapvault.enable off option (turns SnapVault off) 343
snapvault.enable on option (turns SnapVault on) 232, 233, 241, 255, 304, 347, 358
status messages 272
terminology 220
updating schedules snapshots manually 286
updating secondary qtree 246
updating sercondary qtree 285
volume data, starting a backup of 263
snapvault access option 232, 256, 304
SnapVault baseline transfer, definition 221
SnapVault commands
snap sched command (configures primary snapshot schedule) 246, 247, 267
snap sched -x command (configures secondary snapshot schedule) 268
snapvault modify (changes SnapVault configuration) 246, 283
snapvault release (releases unneeded snapshots) 290, 297, 359
snapvault restore (restores data from secondary to primary 236, 289
snapvault snap create (manually updates existing snapshot) 286
snapvault snap unsched command (unconfigures SnapVault) 296
snapvault start (initializes SnapVault backup relationship) 259, 290
snapvault status (checks data replication status) 272
snapvault status (displays information about snapvault snapshots) 272
snapvault stop (ends backup for qtrees) 295
snapvault update (updates SnapVault secondary qtree) 246, 285
SnapVault incremental transfer, definition 221
snapvault modify command (changes SnapVault configuration) 246, 283
SnapVault relationship, definition 220
snapvault release command (releases unneeded snapshots) 290, 297, 359
snapvault restore command (restores from secondary to primary 289
snapvault snap create command (manually updates existing snapshot) 286
snapvault snap sched command (configures primary snapshot schedule) 246, 247, 267
snapvault snap sched -x command (configures secondary snapshot schedule) 268
snapvault snap unsched command (unconfigures SnapVault) 296
snapvault snap -x command supporting Open Systems backup 246
SnapVault snapshot, definition 220
snapvault start command
initializes SnapVault backup relationship 259, 290
supporting Open Systems backup 245
snapvault status command (displays information about snapvault snapshots) 272
snapvault stop command (end backup for qtrees) 295
snapvault update command (updates SnapVault secondary qtree) 246, 285
snapvault.enable off option 343
snapvault.enable on option (turns SnapVault on) 232, 233, 241, 255, 304, 347, 358
Solaris
See also Open Systems
backing up with SnapVault 222
space on disks, how snapshots consume 36
storage systems
creating trust relationships between 384
upgrade order when running SnapMirror 100
synchronous mirroring 79
synchronous SnapMirror 93
cascading from 192
configuration for optimal performance 94

considerations 94
converting from asynchronous 109
filers not supported 94
optimal performance configuration 94
optimized connection 94
replicating data 93
running asynchronously 93
SyncMirror
adding disks to mirrored volumes or aggregates 423
adding synchronous mirroring to an existing traditional volume or aggregate 420
creating mirrored volumes and aggregates 414
removing and destroying a plex 434
removing mirroring between mirrored traditional volumes or aggregates 430
SyncMirror commands
vol add (adds disk to mirrored volume) 424
vol create -m (creates a pair of mirrored volumes) 415
vol destroy (removes a plex from a mirrored pair) 434, 435
vol mirror (adds mirroring to existing volume) 420
vol split command (splits a mirrored volume) 431
vol verify start command (compares data in mirrored volumes) 437
vol verify status command (displays percentage of comparison of data in mirrored volumes completed) 440
vol verify stop command (stops comparing data in mirrored volumes) 439
vol verify suspend command (suspends comparison of data in mirrored volumes) 439
SyncMirror license
disabling 406
enabling 406

T

tape backup of a SnapVault secondary 223
tapes, restarting SnapMirror on multiple 200
TCP window size, globally changing for SnapMirror 103
This 401
traditional volume. *See also* volumes 401
traditional volumes
creating SnapLock 366
transfers, simultaneous, supported per filer 86
troubleshooting reset messages 111
trusted host 384

U

updating SnapMirror mirror 138
upgrade order, storage systems running SnapMirror 100

V

vFilers
controlling use of host filer virus scanners 470
definition 470
virus protection for CIFS 447
virus scanning
adding extensions of files to exclude from a scan 457
adding extensions of files to scan 453
adding shares with virus scanning turned off 463
checking status of vscan options 471
controlling vfiler use of host virus scanners 470
designating secondary virus-scanning clients 451
displaying extensions of files to exclude from scan 456
enables or disables scanning on read-only shares 462
enabling and disabling 451
enabling and disabling mandatory scanning 468
excluding extensions of files to scan 455
extension exclude list compared to extension include list 458
identifying clients that are scanning 465
obtaining Data ONTAP-supported virus-scanning software 448
removing extensions of files to exclude from a

scan 457
removing extensions of files to scan 454, 460
replacing extensions of files to exclude from a scan 456
replacing extensions of files to scan 454
resetting extensions of files to exclude from a scan to default 457
resetting extensions of files to scan to default list 454
resetting scanned files cache 473
resetting the virus scan request timeout 467
setting the virus scan request timeout 467
stopping a scanner session 472
turning scanning off for shares 460
viewing extensions of files to scan 453, 455
vol add command (adds disk to mirrored volume) 424
vol copy throttle command 398
vol create -m command (creates a pair of mirrored volumes) 415
vol create, and SnapLock 214
vol destroy command (removes a plex from a mirrored pair) 434, 435
vol mirror command (adds mirroring to existing volume) 420
vol options commands
vol options nvfail off (disables protection) 444
vol options nvfail on (enables protection) 444
vol restrict command
and SnapLock 214
in SnapMirror 131
vol status -b command 382
vol verify start command (compares data in mirrored volumes) 437
vol verify status command (displays percentage of comparison of data in mirrored volumes completed) 440
vol verify stop command (stops comparing data in mirrored volumes) 439
vol verify suspend command (suspends comparison of data in mirrored volumes) 439
vol.copy.throttle option (controls volume copy speed) 397
volcopy dump operation (backs up to tape) 388
volcopy restore operation (restores backups from tape) 388
volume command, vol volume (changes RAID group size) 407
volume copy
advantages 378
arguments for copying snapshots 389
bringing volumes online (vol online) 385
changing copying speed (options vol.copy.throttle) 397
checking status (vol copy status) 394
copying one volume to another (vol copy start) 391
copying volumes (vol copy start) 388
defined 378
dump operation (reads data from source) 388
enabling Remote Shell access for 387
errors during 392
finding size of volume (vol status -b) 382
finding volume status (online or offline) 385
operation numbers 394
private networks for 380
requirements for 380
restore operation (writes data to destination) 388
sample status message from filer 395
sample status message using rsh 395
stopping (vol copy abort) 399
taking volumes offline (vol offline) 385
to same or different filers 380
uses for 379
volume copy commands
examples of arguments to 391
options rsh.enable on (enables Remote Shell) 387
options vol.copy.throttle (changes copying speed) 397
vol copy abort (halts the copy operation) 399
vol copy arguments for copying snapshots 389
vol copy start (copies one volume to another) 391
vol copy status (checks status) 394
vol offline (takes volume offline) 385
vol online (brings volume online) 385
vol status (shows whether online or offline) 385

vol status -b (finding size of volume) 382
volume copy operation numbers
using with vol copy abort 399
using with vol.copy.throttle 398
volume size, verifying for snapmirror 382
volumes
changing the speed of copying 397
converting from mirror to regular 168
creating traditional SnapLock 366
destination, maximum number of 86
differences between mirror and regular 144
disabling SnapMirror replication 166
enabling SnapMirror replication 128
finding number of blocks in 382
replicating 77
replicating with SnapMirror 82, 128
reverting from snapshots with SnapRestore 68
reverting with SnapRestore 67, 70
See also mirrored volumes and aggregates
vscan commands
cifs shares -add (adds CIFS shares with virus scanning turned off 463
cifs shares -change enabling or disabling scanning for read-only shares 462
cifs shares -change enabling or disabling virus scanning for a share 460
vscan extensions (shows extensions of files to scan) 453, 455
vscan extensions add (adds extensions of files to scan) 453
vscan extensions exclude (excludes extensions of files to scan) 455, 456
vscan extensions exclude add (adds extensions of files to exclude from a scan) 457
vscan extensions exclude remove (removes extensions of files to exclude from a scan) 457
vscan extensions exclude reset (resets extensions of files to exclude from a scan to default list) 457
vscan extensions exclude set (replaces extensions of files to exclude from a scan) 456
vscan extensions remove (removes extensions of files to scan) 454, 460
vscan extensions reset (resets extensions of files to scan to default list) 454
vscan extensions set (replaces extensions of files to scan) 454
vscan off (disables virus scanning) 451
vscan on (enables virus scanning) 451
vscan reset (resets scanned files cache) 473
vscan scanners (identifies clients that are scanning) 465
vscan scanners secondary scanners (designates backup virus scanners) 451
vscan scanners stop (stops scanner session) 472
vscan options
use_host_scanners 470
vscan options mandatory_scan (sets mandatory virus scanning) 468
vscan options timeout reset (resets the virus scan request timeout) 467
vscan options timeout set (sets the virus scan request timeout) 467

W

weekly snapshots 25
Windows 2000
See also Open Systems
backing up with SnapVault 222
Windows NT
See also Open Systems
backing up with SnapVault 222
WORM
data 364
determining if file is 376
requirements 368
transitioning data to 375
WORM volumes. *See* SnapLock