SnapDrive™ for UNIX®
Installation and Administration Guide (IBM® AIX®, HP-UX®, Linux®, Solaris®)

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 210-00875_A0
January 2006

Copyright and trademark information

Copyright information

Copyright © 1994—2006 Network Appliance, Inc. All rights reserved. Printed in the U.S.A.

Term::ReadKey 2.21 - Change terminal modes, and perform non-blocking reads. Copyright (C) 1994 -1999 Kenneth Albanowski, 2001, 2002, Jonathan Stowe. Unlimited distribution and/or modification is allowed as long as this copyright notice remains intact.

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.

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

THIS SOFTWARE IS PROVIDED BY NETWORK APPLIANCE “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 NETWORK APPLIANCE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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

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

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

Trademark information

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

of Network Appliance, Inc. in the United States and other countries. NetApp Availability Assurance and NetApp ProTech Expert are service marks of Network Appliance, Inc. in the United States. Spinnaker Networks, the Spinnaker Networks logo, SpinAccess, SpinCluster, SpinFS, SpinHA, SpinMove, and SpinServer are registered trademarks of Spinnaker Networks, LLC in the United States and/or other countries. SpinAV, SpinManager, SpinMirror, SpinRestore, SpinShot, and SpinStor are trademarks of Spinnaker Networks, LLC in the United States and/or other countries.

Apple is a registered trademark and QuickTime is a trademark of Apple Computer, Inc. in the United States and/or other countries. Microsoft is a registered trademark and Windows Media is a trademark of Microsoft Corporation in the United States and/or other countries. RealAudio, RealNetworks, RealPlayer, RealSystem, RealText, and RealVideo are registered trademarks and RealMedia, RealProxy, and SureStream are trademarks of RealNetworks, Inc. in the United States and/or other countries. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such.

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

Network Appliance NetCache is certified RealSystem compatible.

Table of Contents

Preface ix

Chapter 1 **Overview of SnapDrive for UNIX** 1

How SnapDrive for UNIX works. 2

Comparing SnapDrive for UNIX on different host platforms 10

Where to go for more information 13

Chapter 2 **Installing SnapDrive for UNIX** 15

Verifying that your storage systems are ready 16

Installation instructions for all hosts 21

Installing and uninstalling SnapDrive for UNIX on an AIX host 24

Installing and uninstalling SnapDrive for UNIX on an HP-UX host. 40

Installing and uninstalling SnapDrive for UNIX on a Linux host 47

Installing and uninstalling SnapDrive for UNIX on a Solaris host. 52

Completing the installation 56

Files installed by SnapDrive for UNIX on the host 57

Upgrading to a new version of SnapDrive for UNIX 61

Chapter 3 **Configuring and Using SnapDrive for UNIX** 63

Setting configuration information 64

Preparing hosts for adding LUNs. 88

Setting up audit, recovery, and trace logging. 89

Setting up autosupport 97

General steps for executing commands. 100

Chapter 4 **Setting Up Security Features** 103

Overview of SnapDrive for UNIX's security features 104

Setting up access control 105

View the current access control settings 109
Specifying the current login information for storage systems 111
Enabling SSL encryption 115

Chapter 5 **Creating and Using Snapshots** 117
Creating snapshots 118
Displaying information about snapshots 127
Renaming a snapshot 134
Restoring a snapshot 136
Connecting to a snapshot 147
Disconnecting a snapshot 160
Deleting snapshots 165

Chapter 6 **Provisioning and Managing Storage** 169
Overview of storage provisioning 170
Creating storage 172
Displaying information about storage 185
Increasing the size of the storage 194
Connecting LUNs and storage entities to the host 198
Disconnecting LUN mappings from the host 202
Connecting only the host-side of storage 207
Disconnecting only the host-side of storage 211
Deleting storage from the host and storage system 216

Chapter 7 **Troubleshooting** 219
Data collection utility 220
General troubleshooting tips for SnapDrive for UNIX 223
Understanding error messages 227
Common error conditions 229
Standard exit status values 231

Chapter 8 **Checklists for Using SnapDrive Commands** 245
Collecting information needed by SnapDrive for UNIX Commands 246
Checklist for using the snapshot create command 247
Checklist for using the snapshot rename command 250
Checklist for using the snapshot show command 252
Checklist for using the snapshot restore command 254
Checklist for using the snapshot delete command 257
Checklist for using the snapshot connect command 259
Checklist for using the snapshot disconnect command 263
Checklist for using the storage create command 265
Checklist for using the storage resize command 269
Checklist for using the storage show command 271
Checklist for using the storage connect command 275
Checklist for using the storage disconnect command 277
Checklist for using the host connect command 279
Checklist for using the host disconnect command 281
Checklist for using the storage delete command 284

Chapter 9 **Command Reference** 287
Summary of the SnapDrive for UNIX commands 288
SnapDrive for UNIX keywords and arguments 292

Glossary 305

Index 311

Preface

About this guide

This document describes how to install, configure, and operate SnapDrive™ software for UNIX® servers. It does not cover basic system or network administration topics, such as IP addressing, routing, and network topology. It also does not cover topics that are handled in the Network Appliance™ FCP Host Attach Kit or iSCSI Support Kit documentation. For example, this document does not include information on setting up multipathing.

Audience

This guide is for system administrators who possess working knowledge of NetApp® storage systems, such as filers.

This guide assumes that you are familiar with the following topics:

- Fibre Channel Protocol (FCP)
- iSCSI Protocol
- Basic network functions and operations
- UNIX servers
- UNIX security
- Data storage array administration concepts
- NetApp storage system management
- Logical volume manager on the system you are using

Command conventions

You can enter filer commands on the system console or from any client that can obtain access to the filer using a Telnet session. In examples that illustrate commands executed on a UNIX® workstation, the command syntax and output might differ, depending on your version of UNIX. The examples might be from any of the hosts supported for SnapDrive for UNIX.

Terminology

This guide uses the following terms:

- *LUN* (Logical Unit Number) refers to a logical unit of storage identified by a number.
- *LUN ID* refers to the numerical identifier for a LUN.

- *Storage system* refers to a NetApp® storage system that incorporates the CompactFlash™ unit. Storage systems can support FCP and iSCSI.

  Note

  NetApp's storage products (filers, FAS appliances, NearStore® systems, and storage appliances) are all storage systems. All these products are also sometimes called filers or storage appliances.

- Some UNIX operating systems use the term disk group while others use the term volume group. This guide uses the term *disk group* to refer to both disk groups and volume groups.
- Some operating systems refer to host volumes while others refer to logical volumes. This guide uses the term host volume to refer to both host volumes and logical volumes.

Formatting conventions

The following table lists different character formats used in this guide to set off special information.

Formatting convention	Type of information
Italic type	◆ Words or characters that require special attention. ◆ Placeholders for information you must supply. For example, if the guide requires you to enter the `fctest` *adaptername* command, you enter the characters "fctest" followed by the actual name of the adapter. ◆ 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 your program is case-sensitive and uppercase letters are necessary for it to work properly.

Keyboard conventions

This guide uses capitalization and some abbreviations to refer to the keys on the keyboard. The keys on your keyboard might not be labeled exactly as they are in this guide.

What is in this guide…	What it is used to mean…
hyphen (-)	Separates individual keys. For example, Ctrl-D means holding down the Ctrl key while pressing the D key.
Enter	Refers to the key that generates a carriage return; the key is named Return on some keyboards.
type	Pressing one or more keys on the keyboard.
enter	Pressing one or more keys and then pressing the Enter key.

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.

Overview of SnapDrive for UNIX 1

About this chapter

This chapter provides information about how SnapDrive for UNIX works and what it does.

Topics in this chapter

This chapter includes the following topics:

- "How SnapDrive for UNIX works" on page 2
- "Comparing SnapDrive for UNIX on different host platforms" on page 10
- "Where to go for more information" on page 13

How SnapDrive for UNIX works

What SnapDrive for UNIX does

SnapDrive for UNIX is a tool that simplifies the backup of data so that you can recover it should it be accidentally deleted or modified. SnapDrive for UNIX uses NetApp Snapshot™ technology to create an image (i.e., a snapshot) of the data on a storage system attached to a UNIX host at a specific point in time. If the need arises later, you can restore the data to the storage system. When you restore a snapshot, it replaces the current data on the storage system with the image of the data in the snapshot.

In addition, SnapDrive for UNIX lets you provision storage on the storage system. SnapDrive for UNIX provides a number of storage features that enable you to manage the entire storage hierarchy, from the host-side application-visible file down through the volume manager to the storage-system-side LUNs providing the actual repository.

With SnapDrive for UNIX installed, you can perform the following tasks:

- Create a snapshot of one or more volume groups on a storage system. The snapshot can contain LUNs, file systems, logical volumes, disk groups and NFS directory trees. Once you create a snapshot, you can rename it, restore it, or delete it. You can also connect it to a different location on the host or a different host or disconnect it. In addition, SnapDrive for UNIX lets you display information about snapshots that you created.
- Create storage that includes LUNs, file systems, logical volumes, and disk groups. Then you can resize the storage or delete it. You can also connect the storage to a host or disconnect it. In addition, SnapDrive for UNIX lets you display information about the storage that you created.

Note

SnapDrive for UNIX works only with snapshots it creates. It cannot restore snapshots that it did not create.

How SnapDrive for UNIX creates snapshots

SnapDrive for UNIX software provides commands that you enter on the host that create, restore, and manage snapshots of NetApp storage entities. You can use SnapDrive for UNIX commands to create, restore, and manage snapshots of:

- LVM entities - these are disk groups with host volumes and file systems that you created using the host Logical Volume Manager

- Raw entities - these are either LUNs or LUNs that contain file systems that are mapped directly to the host
- NFS entities - these are NFS files and directory trees

The snapshots that you create can span multiple storage systems and storage system volumes. SnapDrive for UNIX checks the read/writes against the storage entities in the snapshot to ensure that all snapshot data is crash-consistent. SnapDrive will not create a snapshot unless the data is crash-consistent. Refer to "Crash-consistent snapshots" on page 118 for detailed information about crash consistency.

How SnapDrive for UNIX manages storage

SnapDrive for UNIX storage commands help you provision and manage NetApp storage when you create storage entities.

Managing LVM entities: If you request a SnapDrive storage operation that provisions an LVM entity, like a disk group that includes host volumes or file systems, the `snapdrive storage` command works with the LVM to create the LVM objects and file systems that use the storage. During the storage provision operation:

- The host LVM combines LUNs from a storage system into disk or volume groups. This storage is then divided into logical volumes, which are used as if they were raw disk devices to hold file systems or raw data.
- SnapDrive for UNIX integrates with the host LVM to determine which NetApp LUNs make up each disk group, host volume, and file system requested for snapshot. Because data from any given host volume can be distributed across all disks in the disk group, snapshots can be taken and restored only for whole disk groups.

Managing raw entities: If you request a SnapDrive storage operation for a raw entity, like a LUN, or a file system that is created directly on a LUN, SnapDrive performs the storage operation without using the host system LVM. SnapDrive for UNIX allows you to create, delete, connect and disconnect LUNs, and the file systems that they contain, without activating the LVM.

Host communications

SnapDrive for UNIX communicates with the storage system using the host IP interface that you specified when you set up the storage system.

Security considerations

You must be logged in as root to use the SnapDrive for UNIX commands.

To enable SnapDrive for UNIX to access the storage systems connected to the host, you must configure it to use the login names and passwords assigned to the storage systems when you set them up. If you do not provide this information, SnapDrive for UNIX cannot communicate with the storage system. For information on supplying the login names and passwords, see "Specifying the current login information for storage systems" on page 111.

SnapDrive for UNIX stores this information on the host in an encrypted file.

By default, SnapDrive for UNIX does not encrypt the password information when it sends it out across the network. It communicates using HTTP over the standard IP connection. You can tell SnapDrive for UNIX to use HTTPS (SSL encryption) to communicate with the storage system by setting the `use-https-to-filer` variable to On in the snapdrive.conf configuration file. (For more information on the configuration file, see "Setting configuration information" on page 64.).

Note

SnapDrive for UNIX does not use rsh, so you do not need to configure rsh.

Using access permissions on a storage system

SnapDrive for UNIX lets you specify access permissions for each host. These permissions indicate whether a host can perform certain snapshot and storage operations. They do not affect any of the show or list operations.

You set up permissions for each host in a file that resides on the storage system. You can control what action SnapDrive for UNIX takes when it does not find a permissions file for a given host by the value you set in the snapdrive.conf configuration file for the `all-access-if-rbac-unspecified` variable. The options are to allow all access to that storage system or to disable all access to it.

For more information on using access control, see "Setting up access control" on page 105. For more information on the configuration file, see "Setting configuration information" on page 64.

Methods for executing SnapDrive for UNIX

You execute SnapDrive for UNIX from the UNIX host. SnapDrive for UNIX manages snapshots and storage provisioning using a command-line interface. SnapDrive for UNIX lets you:

- Enter individual commands at the command line prompt.
- Run scripts you create that contain embedded SnapDrive for UNIX commands.

Prerequisites for using SnapDrive for UNIX

Prerequisites for SnapDrive for UNIX differ, and depend on whether you have an FCP configuration, an iSCSI configuration, or a configuration that uses NFS directory trees.

FCP or iSCSI Configurations: If you have a configuration that uses FCP or iSCSI, you must do the following before you install SnapDrive for UNIX:

- For FCP configurations, install the Network Appliance™ FCP Host Attach Kit for your host.

  SnapDrive for UNIX works with the following kits:

  - IBM® AIX® FCP attach kit
  - HP-UX® FCP attach kit
  - Solaris® FCP attach kit
- For iSCSI configurations, check that the configuration confirms to requirements defined in the Support Kit documentation for your host.

  SnapDrive for UNIX works with the following support kits:

  - IBM® AIX® iSCSI support kit
  - HP-UX® iSCSI support kit
  - Red Hat® Enterprise Linux® iSCSI support kit
- Set up your host and storage systems.

  Follow the instructions provided with the kit to set up your storage systems to work with the host. SnapDrive for UNIX does not require multipathing or volume management software, however, configurations that include this software must use software that is supported by both the kit and by Snapdrive for UNIX.

Note

The most current information about SnapDrive for UNIX and its requirements is in the *SnapDrive for UNIX Compatibility Matrix* on the NOW site (http://now.netapp.com/NOW/knowledge/docs/docs.shtml).

Configurations that use NFS: For configurations that use NFS you must:

- Check that NFS clients are operating properly. Refer to the *File Access Management Protocols Guide* for detailed information. This document is available at http://now.netapp.com/NOW/knowledge/docs/ontap/rel71rc/. It describes filer operations, and how to manage NFS, CIFS, HTTP, FTP, webDAV and DAFS protocols.
- Set up your host and storage systems.

  To use Snapdrive with NFS-mounted directories on the storage systems, you should ensure that the storage system directories are exported correctly to the host. If your host has multiple IP interfaces that can see the storage

system, ensure that the directory is exported correctly to all of them. Snapdrive will issue warnings unless all such interfaces have read-write permissions, or in the case of the `snap connect` command with the `-readonly` option, at least read-only permissions. The `snap restore` and `snap connect` commands will fail if none of those interfaces have any permissions to access the directory.

SnapDrive for UNIX stack

SnapDrive for UNIX requires the following stack:

- Host operating system and appropriate patches
- Host file systems
  - AIX: JFS2
  - HP-UX: VxFS
  - Linux: Ext3
  - Solaris: VxFS
- Network file systems (NFS)
  - AIX: JFS2
  - HP-UX: VxFS
  - Linux: Ext3
  - Solaris: VxFS
- IP access between the host and storage system
- Storage system licenses
  - FCP, iSCSI, or NFS license, depending on your configuration
  - Flex_clone license (NFS configurations only)
  - SnapRestore® license on the storage system
- Attach or support kit required software

  For example, if you are using SnapDrive for UNIX with an AIX host, you must set up the features required by the attach kit for that host when you are using multipathing. If your configuration uses multipathing, this means you must install the Dot Hill® SANpath® software, which the attach kit uses to handle multipathing.

  Note

  For this release, AIX does not support multipathing if you have the iSCSI support kit installed.

- Data ONTAP® software on your storage system

The most up-to-date information about the requirements for your host operating system are in the online *SnapDrive for UNIX Compatibility Matrix* and the online *Compatibility and Configuration Guide for NetApp's FCP and iSCSI Products.* These matrices are on the NOW site.

Supported FCP/iSCSI or NFS configurations

SnapDrive for UNIX supports following host cluster and storage cluster topologies:

- Nonclustered configuration where a single host is connected to a single storage system
- Any of the topologies involving NetApp storage system cluster failover
- Any of the topologies involving NetApp-supported host clusters

FCP/iSCSI configurations support the same host cluster and storage system cluster configurations that the attach or support kit supports. See the kit documentation for more information about the recommended configurations for your host and the storage systems you are using.

Note

If you need a SnapDrive for UNIX configuration that is not mentioned in the kit documentation, consult with your NetApp sales representative.

Architectural guidelines for SnapDrive for UNIX

The following are architectural guidelines for this release of SnapDrive for UNIX:

- A LUN managed by SnapDrive for UNIX cannot serve as either of the following:
  - A boot disk or a system disk
  - A location for the system paging file or memory dump files (or swap disk)
- Snapshot support for storage entities spanning multiple storage system volumes or multiple storage systems is limited on configurations that do not permit a freeze operation in the software stack. Refer to "Crash-consistent snapshots" on page 118 for additional information.
- Support for volume groups spanning multiple storage system volumes or multiple storage systems is limited. In particular, SnapDrive for UNIX has the following limitation:
  - You cannot create volume groups that span storage systems using the `storage create` command.

  Four key commands that SnapDrive for UNIX supports in these cases are:

- The `snap create` command
- The `snap restore` command
- The `snap connect` command
- The `snap disconnect` command

- On HP-UX hosts, the default behavior of the LVM is to limit the host to 16 physical volumes per host volume group. This limits the maximum number of LUNs to 16. This limit affects the `snapdrive storage create` and the `snapdrive storage resize` commands. If you attempt to create a host volume with more than 16 LUNs, the `snapdrive storage create` command fails. If you try to resize a volume group so that it contains more than 16 LUNs, the snapdrive `storage resize` command fails.

  To have more than 16 LUNs (physical volumes) on an HP-UX host, you must manually create the host volume group using the `vgcreate` command and set the value for `-p max_pv` to the maximum value you expect to need for LUNs. The only time you may set this limit is when you create the host volume group. See the HP-UX `vgcreate`(1) man page for more information,
- Connecting to the originating host occurs when you use `snap connect` to connect a snapshot at a new location on the same host where it was last connected (or is still connected).

  On Linux hosts, the `snap connect` command is supported providing the snapshot you connect contains a LUN, or a LUN with a file system, that was created without activating the Linux LVM. SnapDrive for UNIX does not support the `snap connect` command for Linux entities that are created using the Linux LVM. Refer to "Connecting to a snapshot" on page 147 for additional information.
- The following limitations apply to SnapDrive for UNIX operations that involve NFS files or directory trees:
  - SnapDrive for UNIX does not provide storage provisioning commands for NFS files or directory trees.
  - SnapDrive for UNIX supports the `snapdrive snap create` and `snapdrive snap restore` commands on versions of Data ONTAP 6.5 and above. However, `snapdrive snap connect` and `snapdrive snap disconnect` commands that involve NFS use the Data ONTAP FlexVol volumes feature for read and write access, and therefore require Data ONTAP 7.0 and FlexVol volumes. Configurations with Data ONTAP 6.5, or Data ONTAP 7.0 and traditional volumes can create and restore snapshots, but `snap connect` operations are restricted to read only access.
  - The NFS file or directory tree that you specify using the `snapdrive snap create` command must include a entry in the file system table file.

SnapDrive will not create a snapshot of NFS entities that do not have an mount entry. The SnapDrive for UNIX `snapdrive snap restore` command automatically restores and mounts the NFS file or directory tree that you specify. You can not use the `-nopersist` option to remove mount entries from the file system mount table.

- The following guidelines apply to the `storage create` command:
  - If you use the `storage create` command to create a logical volume or a file system on a LUN, you cannot directly specify the size of the logical volume or file system being created. Instead, you must specify the size of the underlying volume group; any host volume or file system created within it will be marginally smaller than it is.
- If you want to resize your storage, you must use the `-addlun` option on the `storage resize` command line. SnapDrive for UNIX does not currently support expanding existing LUNs.
- The `storage resize` command does not work with LUNs mapped directly to the host, or with the files systems that they contain. Refer to "Increasing the size of the storage" on page 195 for additional information.
- In creating host volumes, SnapDrive for UNIX does not provide any options to control their formatting. It creates only concatenated host volumes. SnapDrive for UNIX does operate correctly on host volumes of other formats (such as striped volumes) that were created outside of it.
- When you create a snapshot on a mirrored filer, the snapshot is automatically repilcated, from the source filer where it is created, to the destination (mirrored) filer. The previous version of SnapDrive for UNIX allowed you to restore the snapshot only on the source filer. This version of SnapDrive for UNIX allows you to restore the snapshot on the destination filer as well. Refer to "Restoring snapshots on mirrored filers" on page 137 for additional information.
- You cannot restore a portion of a disk group. SnapDrive for UNIX backs up and restores whole disk groups only.
- Solaris and Linux hosts have operating system limits on how many LUNs you can create. To avoid having a problem when you create LUNs on these hosts, use the `snapdrive config check luns` command. For more information, see "Preparing hosts for adding LUNs" on page 88.
- SnapDrive for UNIX does not support colons in the long forms of the names for LUNs and snapshots. The only place SnapDrive for UNIX accepts colons is between the components of a long snapshot name or between the storage system name and the storage system volume name of a LUN. For example, `toaster:/vol/vol1:snap1` would be a typical long snapshot name, while `toaster:/vol/vol1/lunA` would be a typical long LUN name.

Comparing SnapDrive for UNIX on different host platforms

About SnapDrive for UNIX on multiple platforms

In the majority of ways, SnapDrive for UNIX executes the same way on multiple host operating system platforms. This section describes the differences in how SnapDrive for UNIX executes across host operating systems. Except for these points, this document describes how to use SnapDrive for UNIX regardless of the host operating system. Differences in SnapDrive for UNIX behavior between the operating systems include the following:

- Installation instructions
- Pathnames
- Terminology
- Output from examples
- Cross-volume snapshot support
- Originating-host support of `snap connect` operation

This document contains separate installation instructions for each operating system. In addition, it provides a list of files that each operating system installs so you can see the pathnames used by that operating system.

In an effort to make this document easier to read, it uses one term for an item regardless of the operating system. For example, the terms *disk group* and *volume group* in this documentation can refer to either a disk group or a volume group. The term *host volume* refers to either a host volume or a logical volume; both are distinguished from storage system volumes.

Wherever possible, this document uses generic examples. In some cases, this document provides multiple examples to show the output from different operating systems. In most cases, the differences in output have to do with the differences in pathnames and terminology on an operating system so they are clear to users familiar with UNIX operating systems. For example, an HP-UX host uses /dev/rdsk/c22t0d0 as a device name while a Solaris host uses /dev/vx/rdmp/c1t0d2s2, an AIX host uses /dev/hdisk2, and a Linux host uses /dev/sdd. To avoid making the examples overly long, this document sometimes uses one operating system for one example and a different operating system for the next example.

Differences between host platforms

There are three main differences in supported SnapDrive for UNIX functionality across different host operating systems

- Snapshot support for storage entities spanning multiple storage system volumes or multiple storage systems is limited on hosts or configurations that do not permit a freeze operation in the software stack. Refer to "Crash-consistent snapshots" on page 118 for additional information.
- Using the `snap connect` command to connect LUNs to the originating host is supported on AIX, HP-UX, and Solaris hosts.

  On Linux hosts, using the `snap connect` command to connect to the originating host is supported providing the snapshot you connect contains a LUN, or a LUN with a file system, that was created without activating the Linux LVM. SnapDrive for UNIX does not support the `snap connect` command for Linux entities that are created using the Linux LVM.
- You must prepare Linux and Solaris hosts before you add LUNs. See "Preparing hosts for adding LUNs" on page 88.

For more information on these differences, see "Architectural guidelines for SnapDrive for UNIX" on page 7).

SnapDrive for UNIX terms for volume managers

The following table summarizes some of the differences in terms when referring to volume managers on the host platforms.

Host	Volume manager	Volume or disk groups	Location of logical volumes	Location of multipathing devices
AIX	Native Logical Volume Manager (LVM)	Volume groups (*vg*)	/dev/*lvol-name* Note: All logical volumes share the same namespace.	/dev/hdisk (FCP only) Multipathing not supported with iSCSI
HP-UX	Native Logical Volume Manager (LVM)	Volume groups (*vg*)	/dev/*dg-name* /*volgroup-name*	/dev/rdsk
Linux	Native Logical Volume Manager (LVM)	Volume groups (*vg*)	/dev/*dg-name* /*lvol-name*	Multipathing not supported

Host	Volume manager	Volume or disk groups	Location of logical volumes	Location of multipathing devices
Solaris	Veritas Volume Manager (VxVM)	Disk groups (*dg*)	/dev/vx/*dg-name*/*lvol-name*	/dev/vx/rdmp

Where to go for more information

Relevant documentation

This guide provides information about the basic tasks involved in installing SnapDrive for UNIX on your host and working with snapshots. The following documentation might also be useful to you:

- *SnapDrive for UNIX Release Notes (IBM AIX, HP-UX, Linux, Solaris)*

  This document comes with SnapDrive for UNIX. You can also download a copy from http://now.netapp.com. It contains any last-minute information that you need to get your configuration up and running smoothly. It also contains late-breaking problems and their workarounds.

- *SnapDrive for UNIX Quick Start Guide (IBM AIX, HP-UX, Linux, Solaris)*

  This document comes with SnapDrive for UNIX. You can also download a copy from http://now.netapp.com. It provides high-level steps so that you can quickly start using SnapDrive for UNIX to create snapshots and manage storage.

- *SnapDrive for UNIX Compatibility Matrix*

  This document is available at http://now.netapp.com/NOW/knowledge/docs/docs.shtml. It is a dynamic, online document that contains the most up-to-date information specific to SnapDrive for UNIX and its platform requirements.

- *SnapDrive for UNIX* man page

  This online document comes with the product. It contains descriptions of the SnapDrive for UNIX commands and covers issues such as using initiator groups and internal name generation.

- *Compatibility and Configuration Guide for NetApp's FCP and iSCSI Products*

  This document is available at http://now.netapp.com/NOW/knowledge/docs/san/fcp_iscsi_config/. It is a dynamic, online document that contains the most up-to-date information about the requirements for setting up a system in a NetApp SAN environment. It provides the most current details about storage systems and host platforms, cabling issues, switch issues, and configurations.

- *File Access Management Protocols Guide*

  This document is available at http://now.netapp.com/NOW/knowledge/docs/ontap/rel71rc/. It describes filer operations, and how to manage NFS, CIFS, HTTP, FTP, webDAV and DAFS protocols.

- The FCP host attach kit documentation

  This documentation comes with the attach kit. It consists of the installation guide for your host and the release notes for that attach kit.
- The NetApp iSCSI host support kit documentation

  This documentation comes with the support kit. It consists of the installation guide for your host and the release notes for that support kit.
- *System Configuration Guide*

  This document is available at http://now.netapp.com/NOW/knowledge/docs/docs.shtml/. This is an online document that contains information about the supported storage system models for Data ONTAP.
- *Data ONTAP Block Access Management Guide*

  This document is available from the Data ONTAP library on the NOW site at http://now.netapp.comNOW/knowledge/docs.shtml/. It provides information about using Data ONTAP and setting up your storage system to work with Data ONTAP.
- Host operating system and host bus adapter (HBA) information

  NetApp does not provide these documents. See the readme files and other documentation that you received with your host operating system.

Installing SnapDrive for UNIX 2

About this chapter

This chapter explains the procedures you must follow to install SnapDrive for UNIX.

Topics in this chapter

This chapter discusses the following topics:

- "Verifying that your storage systems are ready" on page 16
- "Installation instructions for all hosts" on page 21
- "Installing and uninstalling SnapDrive for UNIX on an AIX host" on page 24
- "Installing and uninstalling SnapDrive for UNIX on an HP-UX host" on page 40
- "Installing and uninstalling SnapDrive for UNIX on a Linux host" on page 47
- "Installing and uninstalling SnapDrive for UNIX on a Solaris host" on page 52'
- "Completing the installation" on page 56
- "Files installed by SnapDrive for UNIX on the host" on page 57
- "Upgrading to a new version of SnapDrive for UNIX" on page 61

Verifying that your storage systems are ready

Verify storage system readiness

Before installing SnapDrive for UNIX, verify that your the storage systems are ready by making sure that the following conditions are true:

- The storage systems are online.
- The storage systems meet the minimum system requirements for SnapDrive for UNIX. See "Requirements for storage systems" on page 16.
- The HBAs and/or network interface cards (NICs) in your storage systems meet the requirements for your host operating system.
- The hosts and the storage systems can communicate using an IP interface. (You should have set this up when you set up your storage system.)
- SnapRestore is licensed. (You should have set this up when you set up your storage system.)

For detailed information about administering your storage system, see your Data ONTAP *Storage Management Guide*.

Requirements for storage systems

Each storage system in your SnapDrive for UNIX configuration must meet the requirements in the following table.

Component	Minimum requirement
Operating system	Data ONTAP 6.5.2 or later ◆ SnapDrive for UNIX supports FlexVol volumes, but does not take advantage of all FlexVol volume features ◆ Configurations that use NFS must use Data ONTAP 7.0 or greater and FlexVol volumes to use `snap connect` to read and write to a connected NFS file or directory tree. Configurations with traditional volumes are provided with read only access to NFS files and directory trees.

Component	Minimum requirement
storage system setup	You must specify the partner IP address in the storage system cluster that can be used if a storage system failover occurs. **Note** You specify this address when you run the setup program on the storage system. See "Confirm storage system has partner IP address" on page 17.
Licenses	◆ FCP, iSCSI, or NFS, depending on your host platform ◆ NFS configurations must have a FlexClone license **Note** You must have the correct protocol running on the storage system for SnapDrive for UNIX to execute. ◆ SnapRestore software

Note
For the latest SnapDrive for UNIX requirements, see the online *SnapDrive for UNIX Compatibility Matrix* and the *Compatibility and Configuration Guide for NetApp's FCP and iSCSI Products*. These matrices are at http://now.netapp.com/NOW/knowledge/docs/docs.shtml/.

Confirm storage system has partner IP address

When you ran the setup program on your storage system, it prompted you for an IP address for a partner storage system to use in case of a failover. Make sure that you supplied this address. If you didn't supply it, SnapDrive for UNIX cannot inquire about the storage entities on a storage system that was taken over.

Example: The following is the portion of the storage system setup script that requests the IP address. This example uses the IP address 10.2.21.35.

```
filer_A> setup
...
Should interface e0 take over a partner IP address during failover?
[n]: y
```

```
Please enter the IP address or interface name to be taken over by e0
[]: 10.2.21.35
...
reboot -t 0
```

NFS considerations

The following considerations apply to configurations that use SnapDrive in an NFS environment:

- The NFS service must be running on the storage system.
- The NFS client must have permissions to export (mount) resources from the storage system.

  If you are using Snapdrive to restore or connect to NFS-mounted directories on the storage systems, you should ensure that the storage system directories are exported correctly to the host. If your host has multiple IP interfaces that can access the storage system, ensure that the directory is exported correctly to all of them. Snapdrive will issue warnings unless all such interfaces have read-write permissions, or in the case of `snap connect` with the `-readonly` option, at least read-only permissions. The `snap restore` and `snap connect` commands will fail if none of these interfaces have any permissions to access the directory.

Refer to the *File Access Management Protocols Guide* for detailed information. This document is available at http://now.netapp.com/NOW/knowledge/docs/ontap/rel71rc/. It describes filer operations, and how to manage NFS, CIFS, HTTP, FTP, webDAV and DAFS protocols.

Cautions for using SnapDrive for UNIX

NetApp strongly recommends that you heed the following cautions:

- Use the default value for the space reservation setting for any LUN managed by SnapDrive.
- In FCP or iSCSI configurations, set the snap reserve on the storage system to 0 percent for each volume.
- Place all LUNs connected to the same host on a dedicated storage system volume accessible by just that host.
- If you use snapshots, you cannot use the entire space on a storage system volume to store your LUNs.

  The storage system volume hosting the LUNs should be at least twice the combined size of all the LUNs on the storage system volume.
- Data ONTAP uses /vol/vol0 to administer the storage system. Do not use this volume to store data.

Preparing a storage system volume

You need to perform the following tasks on the storage system to create a volume that can hold the SnapDrive for UNIX LUNs or NFS entities attached to a single host:

- Create a storage system volume.

  Note

  You can use either the command-line prompt on the storage system (for example, through a Telnet session) or FilerView to create a storage system volume dedicated to SnapDrive for UNIX.

  For more information about the following procedures, see the Data ONTAP *Block Access Management Guide*.

- If you are in an FCP or iSCSI environment, reset the snap reserve option to 0 percent on the storage system volume holding all the LUNs attached to the host (optional, but highly recommended). Refer to "Resetting the snap reserve option" on page 19.

When you create a volume on a storage system to hold LUNs or NFS directory trees, keep the following in mind:

- You can create multiple LUNs or NFS directory trees on a storage system volume.
- You can not store user data in the root storage system volume (for example, /vol/vol0).

Optimizing storage system volumes in a FCP or iSCSI environment:

You can optimize your storage system volumes in the following ways:

- When multiple hosts share the same storage system, each host should have its own dedicated storage system volume to hold all the LUNs connected to that host.
- NetApp recommends that, when multiple LUNs exist on a storage system volume, the dedicated volume on which the LUNs reside contains only the LUNs for a single host. It must not contain any other files or directories.

Resetting the snap reserve option

By default, the `snap reserve` option for Data ONTAP 6.5.*x* is 20 percent. When you use Data ONTAP in an FCP or iSCSI environment, NetApp strongly recommends that you reset the `snap reserve` option to 0 percent on all storage system volumes holding SnapDrive for UNIX LUNs.

To reset the `snap reserve` option on the storage system, complete the following steps.

Step	Action
1	Access the storage system either by using a command such as `telnet` from the host or by going to the storage system console.
2	Enter the following command: `# snap reserve vol_name 0` *vol_name* is the name of the volume where you want to set the `snap reserve` option.

To reset the `snap reserve` option using FilerView, complete the following steps.

Step	Action
1	Open a FilerView session to the storage system holding the volume whose `snap reserve` setting is to be changed.
2	From the main FilerView menu, navigate to Volumes > Snapshots > Configure.
3	In the Volume field, select the volume whose snap reserve setting is to be changed
4	In the space reservation field, enter “0.”
5	Click Apply at the bottom of the panel.

Installation instructions for all hosts

About installing the attach kit

Before you install SnapDrive for UNIX, you must ensure that your host and storage system are configured properly.

- If your configuration requires an attach kit, you must install it and get it working.
- If your configuration uses a support kit, you must consult support kit documentation to ensure that the system is set up properly.

Use the documentation that came with the attach or support kit. It contains information about volume managers, multipathing, and other features you need to set up before you install SnapDrive.

Verify that the hosts are ready

Verify that the hosts are ready by performing the following tasks:

- Confirm that the host and storage system can communicate.

  Confirm that you set up the host and storage system correctly according to the instructions in the attach kit for the host.
- If you have a configuration that uses NFS, configure the exports file. Refer to the *File Access and Protocols Management Guide* on http://now.netapp.com/NOW/knowledge/docs/ontap/ontap_index.shtml for additional information.
- Verify that the host meets the minimum requirements for SnapDrive for UNIX, including the required operating system patches.

Note

For the latest information about the SnapDrive for UNIX requirements and the attach kits that NetApp provides, see the online *SnapDrive for UNIX Compatibility Matrix* and the *Compatibility and Configuration Guide for NetApp's FCP and iSCSI Products.* These matrices are at http://now.netapp.com/NOW/knowledge/docs/docs.shtml/.

Get a copy of the SnapDrive for UNIX software package

You can obtain the SnapDrive for UNIX software package in two ways:

- Download the package from http://now.netapp.com. The SnapDrive for UNIX software is bundled into a single compressed file. For information

about getting this file, see “Downloading the SnapDrive for UNIX software from NOW” on page 22.

- Get the package from the SnapDrive for UNIX CD-ROM. For information about getting the software package from the CD-ROM, see“Getting SnapDrive for UNIX software from the CD-ROM” on page 23.

Note

To verify that you have the latest software package, check the online *SnapDrive for UNIX Compatibility Matrix* and the *Compatibility* and the *Configuration Guide for NetApp's FCP and iSCSI Products*. They are both available at http://now.netapp.com/NOW/knowledge/docs/docs.shtml.

Downloading the SnapDrive for UNIX software from NOW

To download the SnapDrive for UNIX software from NOW, complete the following steps.

Step	Action
1	Log in to http://now.netapp.com. In the fields on the right side of the page, complete the following steps: **1.** Enter your user name and password. **2.** From the Select Start Page list box, click Download Software. **3.** Click the Login button.
2	Go to the SnapDrive for UNIX product row of the Software Download table and select your host operating system from the Select Platform drop-down list.
3	Click Go!
4	Follow the prompts to reach the Software Download page.
5	Download the software file to a local directory. If you did not download the file to your host machine, move it to that machine.
6	Go to the instructions for installing the SnapDrive for UNIX software on your host operating system.

Getting SnapDrive for UNIX software from the CD-ROM

To get the SnapDrive for UNIX software from the CD-ROM, complete the following steps.

Step	Action
1	Insert the CD-ROM containing the version of SnapDrive for UNIX for your host operating system into the CD-ROM drive.
2	Change to the directory containing the SnapDrive for UNIX software.

Installing and uninstalling SnapDrive for UNIX on an AIX host

System requirements for FCP/iSCSI configurations

The following table lists the minimum requirements for using SnapDrive with an AIX host in an FCP or iSCSI environment.

Component	Requirement
NetApp iSCSI Support Kit or FCP AIX Attach Kit	To make sure you have the correct version of the kit, see the online *SnapDrive for UNIX Compatibility Matrix* and the *Compatibility and Configuration Guide for NetApp's FCP and iSCSI Products.* Set up the host and storage system according to the instructions in the installation and setup guide for the iSCSI or FCP kit. You must do this before you install SnapDrive for UNIX.
Additional disk space	SnapDrive for UNIX maintains three log files. While SnapDrive for UNIX rotates the files when they reach a maximum size, you should make sure you have sufficient disk space for them. For more information about the log files, see "Setting up audit, recovery, and trace logging" on page 89. Based on the default settings for the audit and trace log files, you need at least 1.1 MB of space. There is no default size for the recovery log because it rotates only after an operation completes, not when it reaches a specific size.

Getting the software package

You can obtain the SnapDrive for UNIX software package in two ways:

- Download the package from http://now.netapp.com. See "Downloading the SnapDrive for UNIX software from NOW" on page 22.
- Get the package from the SnapDrive for UNIX CD-ROM provided with the media package for your host operating system. See "Getting SnapDrive for UNIX software from the CD-ROM" on page 23.

Uncompress the software if you downloaded it

If you downloaded the software package from NOW, you can uncompress it by completing the following steps:

Step	Action
1	Uncompress the file and extract the software by entering the following command: `# uncompress NetApp.snapdrive_aix_2_1.Z` **Result:** This command extracts the installation package NetApp.snapdrive_aix_2_1.
2	You're now ready to install SnapDrive for UNIX. Follow the steps in the section "Installing SnapDrive for UNIX on an AIX host" on page 25.

Installing SnapDrive for UNIX on an AIX host

You use the System Management Interface Tool (SMIT) to install the AIX package. Complete the following steps.

Step	Action
1	Make sure you are logged in as root.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="3">2</td><td>If you are installing...</td><td>Then...</td></tr>
<tr><td>Software from the CD-ROM</td><td>1. Insert the CD-ROM into the CD-ROM drive.
2. Change to the root directory, which contains the software package NetApp.snapdrive_aix_2_1.
Note
This package is in a standard AIX Licensed Program Product (LPP) format.
3. Optional: Copy the package from the CD-ROM's root directory to a temporary location on the host. (You can also install the package from the CD-ROM.)
Example: If you mount the CD-ROM on /mnt and want to place the package in the /tmp directory, you might enter commands similar to the following:
<code># mount -v cdrfs -o ro /dev/cd0 /mnt
cp /mnt/install /tmp
cd /tmp</code></td></tr>
<tr><td>Software from NOW</td><td>Change to the directory where you placed the uncompressed file NetApp.snapdrive_aix_2_1.</td></tr>
<tr><td>3</td><td colspan="2">Start SMIT by entering the following command:
<code># smit</code></td></tr>
</table>

Step	Action
4	Select the option titled “Software Installation and Maintenance.” **Example:** When you start SMIT, it displays the following screen. On this screen, the “Software Installation and Maintenance” option is the first menu option. +--+ root> # smit System Management Move cursor to desired item and press Enter. Software Installation and Maintenance Software License Management Devices System Storage Management (Physical & Logical Storage) Security & Users Communications Applications and Services Print Spooling Problem Determination Performance & Resource Scheduling System Environments Processes & Subsystems Applications Cluster System Management Using SMIT (information only) F1=Help F2=Refresh F3=Cancel Esc+8=Image Esc+9=Shell Esc+0=Exit Enter=Do +--+

Step	Action
5	At the screen that appears, select the Install and Update Software menu option. **Example:** The following is an example of the Software Installation and Maintenance screen. +--+ Software Installation and Maintenance Move cursor to desired item and press Enter. Install and Update Software List Software and Related Information Software Maintenance and Utilities Network Installation Management System Backup Manager F1=Help F2=Refresh F3=Cancel Esc+8=Image Esc+9=Shell Esc+0=Exit Enter=Do +--+
6	At the next screen, select the Install Software menu option. **Example:** The following is an example of the Install and Update Software screen. +--+ Install and Update Software Move cursor to desired item and press Enter. Install Software Update Installed Software to Latest Level (Update All) Install Software Bundle Update Software by Fix (APAR) Install and Update from ALL Available Software F1=Help F2=Refresh F3=Cancel Esc+8=Image Esc+9=Shell Esc+0=Exit Enter=Do +--+

Step	Action

7	At the Install Software screen, specify the location of the software in one of the following ways:
	◆ Manually enter the location by providing the following information.
	❖ If you're installing from the CD-ROM, enter the CD-ROM drive.
	❖ If you're installing from the host machine, enter the path to the software package (for example, /tmp/NetApp.snapdrive_aix_2_1).
	◆ Press F4 to display a list of options.
	If you want to use the F4 method, complete the following steps:
	1. Press F4.
	2. At the prompt for software to install, enter NetApp.snapdrive.
	3. At the prompt asking whether you want to continue or cancel, press Enter to complete the installation.
	The installation process checks the version of AIX. The installation process displays a warning message before it completes if you are not running a version of AIX that SnapDrive for UNIX supports:
	Example 1: The following is an example of entering the path to the software package when you're at the Install Software screen.
	`+--------------------------------------------------------------------------+`
	`Install Software`
	
	`Type or select a value for the entry field.`
	`Press Enter AFTER making all desired changes.`
	
	`[Entry Fields]`
	`* INPUT device / directory for software [/tmp/aix/NetApp.snapdrive_aix_2_1]+`
	
	`F1=Help F2=Refresh F3=Cancel F4=List`
	`Esc+5=Reset Esc+6=Command Esc+7=Edit`
	`Esc+8=Image`
	`Esc+9=Shell Esc+0=Exit Enter=Do`
	`+--------------------------------------------------------------------------+`

Step	Action
	Example 2: After you enter path to the software package, SMIT displays the following screen. This is the screen where you enter name of the software package, NetApp.snapdrive.

```
+--------------------------------------------------------------------------+
Install Software

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* INPUT device / directory for software
/tmp/aix/NetApp.snapdrive_aix_2_1
* SOFTWARE to install                                [NetApp.snapdrive]       +
  PREVIEW only? (install operation will NOT occur)    no                     +
  COMMIT software updates?                            yes                    +
  SAVE replaced files?                                no                     +
  AUTOMATICALLY install requisite software?           yes                    +
  EXTEND file systems if space needed?                yes                    +
  OVERWRITE same or newer versions?                   no                     +
  VERIFY install and check file sizes?                no                     +
  Include corresponding LANGUAGE filesets?            yes                    +
  DETAILED output?                                    no                     +
  Process multiple volumes?                           yes                    +
  ACCEPT new license agreements?                      no                     +
  Preview new LICENSE agreements?                     no                     +



F1=Help                 F2=Refresh              F3=Cancel               F4=List
Esc+5=Reset              Esc+6=Command            Esc+7=Edit
Esc+8=Image
Esc+9=Shell              Esc+0=Exit               Enter=Do
+--------------------------------------------------------------------------+
```

Step	Action

	Example of a successful installation: The following is an example of output you might see when an installation successfully completes.
	
	`File:`
	`    I:NetApp.snapdrive             2.1`
	
	`+-------------------------------------------------------------------------------+`
	`                    Pre-installation Verification...`
	`+-------------------------------------------------------------------------------+`
	`Verifying selections...done`
	`Verifying requisites...done`
	`Results...`
	
	`SUCCESSES`
	`---------`
	`  Filesets listed in this section passed pre-installation verification`
	`  and will be installed.`
	
	`  Selected Filesets`
	`  -----------------`
	`  NetApp.snapdrive 2.1 # Network Appliance Snapdrive`
	
	`  << End of Success Section >>`
	
	`FILESET STATISTICS`
	`------------------`
	`    1  Selected to be installed, of which:`
	`        1  Passed pre-installation verification`
	`  ----`
	`    1  Total to be installed`

Step	Action
	+--+ Installing Software... +--+ installp: APPLYING software for: NetApp.snapdrive 2.1 . . . . . << Copyright notice for NetApp.snapdrive >> . . . . . . Copyright (c) 2004 Network Appliance, Inc. All Rights Reserved. . . . . . << End of copyright notice for NetApp.snapdrive >>. . . . Finished processing all filesets. (Total time: 2 secs). +--+ Summaries: +--+ Installation Summary -------------------- Name Level Part Event Result --- NetApp.snapdrive 2.1 USR APPLY SUCCESS
8	Verify your installation of the package by entering the `lslpp -l NetAppsnapdrive` command line. It should produce the following output: # lslpp -l NetApp.snapdrive Fileset Level State Description -- Path: /usr/lib/objrepos NetApp.snapdrive 2.1 COMMITTED Network Appliance Snapdrive You can also check the SMIT log file (smit.log and smit.script). These files are in the SMIT log directory ($HOME).

Step	Action
9	Complete your setup by configuring SnapDrive for UNIX for your system. Most of this information is set by default; however, you need to specify the following information: ◆ The login information for the storage system. See "Specifying the current login information for storage systems" on page 111. ◆ The autosupport settings (autosupport is an optional feature, but NetApp recommends you enable it). See "Setting up autosupport" on page 97. ◆ The correct configuration value for the following options based on whether you are using the FCP protocol or the iSCSI protocol: ❖ default-transport. Set this equal to your protocol. The acceptable values are "fcp" and "iscsi". ❖ multipathing-type. For FCP, set this to "SANPath". For iSCSI, set this to "none". Multipathing is not available with iSCSI for this release. **Note** For general information about configuration settings, see "Setting configuration information" on page 64.

Uninstalling the software from an AIX host

You use SMIT to uninstall SnapDrive for UNIX from an AIX system. Complete the following steps.

Step	Action
1	Ensure that you are logged in as root.
2	Start SMIT by entering the following command: `# smit`

Step	Action
3	Select the “Software Installation and Maintenance” menu option. **Example:** When you start SMIT, it displays the following screen. +--+ root> # smit System Management Move cursor to desired item and press Enter. Software Installation and Maintenance Software License Management Devices System Storage Management (Physical & Logical Storage) Security & Users Communications Applications and Services Print Spooling Problem Determination Performance & Resource Scheduling System Environments Processes & Subsystems Applications Cluster System Management Using SMIT (information only) F1=Help F2=Refresh F3=Cancel Esc+8=Image Esc+9=Shell Esc+0=Exit Enter=Do +--+

Step	Action
4	At the screen that appears, select the Software Maintenance and Utilities menu option. **Example:** The following is an example of the Software Installation and Maintenance screen. +--+ Software Installation and Maintenance Move cursor to desired item and press Enter. Install and Update Software List Software and Related Information Software Maintenance and Utilities Network Installation Management System Backup Manager F1=Help F2=Refresh F3=Cancel Esc+8=Image Esc+9=Shell Esc+0=Exit Enter=Do +--+

Step	Action

5	At the next screen, select the Remove Installed Software menu option.
	
	Example: The following is an example of the Software Maintenance and Utilities screen.
	
	`+-------------------------------------------------------------------------------+`
	`Software Maintenance and Utilities`
	
	`Move cursor to desired item and press Enter.`
	
	`  Commit Applied Software Updates (Remove Saved Files)`
	`  Reject Applied Software Updates (Use Previous Version)`
	`  Remove Installed Software`
	
	`  Copy Software to Hard Disk for Future Installation`
	
	`  Check Software File Sizes After Installation`
	`  Verify Software Installation and Requisites`
	
	`  Clean Up After Failed or Interrupted Installation`
	
	`F1=Help  F2=Refresh  F3=Cancel`
	`Esc+8=Image`
	`Esc+9=Shell  Esc+0=Exit  Enter=Do`
	`+-------------------------------------------------------------------------------+`

Step	Action
6	Remove the software in one of the following ways: ◆ Enter the package name (NetApp.snapdrive). (Make sure that the "Preview only" option is set to no.) ◆ Press F4 to display a list of names. If you use F4, complete the following steps: **1.** Press F4. **2.** Scroll down the list of names until you reach NetApp.snapdrive. **3.** Select NetApp.snapdrive and press Enter. **4.** At the prompt asking whether you want to continue or cancel, press Enter to complete the uninstall. The uninstall process places the log file in /tmp/snapdrive_uninstall. **Note** When you unstall the software, the log files are not removed. You must go to the /var/snapdrive directory and manually remove them. **Example:** The following is an example of the Remove Installed Software screen. **Note** By default PREVIEW only? is set to yes. You must change it to no if you want to uninstall.

```
+--------------------------------------------------------------------------+
Remove Installed Software
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* SOFTWARE name                                     [NetApp.snapdrive]        +
  PREVIEW only? (remove operation will NOT occur)     no                      +
  REMOVE dependent software?                          no                      +
  EXTEND file systems if space needed?                no                      +
  DETAILED output?                                    no                      +


F1=Help                 F2=Refresh              F3=Cancel               F4=List
Esc+5=Reset              Esc+6=Command            Esc+7=Edit
Esc+8=Image
Esc+9=Shell              Esc+0=Exit               Enter=Do
+--------------------------------------------------------------------------+
```

Step	Action

	Example of successful uninstall: The following output appears when you successfully uninstall the software.
	
	`File:`
	`NetApp.snapdrive`
	
	`+-------------------------------------------------------------------------------+`
	`Pre-deinstall Verification...`
	`+-------------------------------------------------------------------------------+`
	`Verifying selections...done`
	`Verifying requisites...done`
	`Results...`
	
	`SUCCESSES`
	`---------`
	`Filesets listed in this section passed pre-deinstall verification`
	`and will be removed.`
	
	`Selected Filesets`
	`-----------------`
	`NetApp.snapdrive 2.1 # Network Appliance Snapdrive`
	
	`<< End of Success Section >>`
	
	`FILESET STATISTICS`
	`------------------`
	`1 Selected to be deinstalled, of which:`
	`1 Passed pre-deinstall verification`
	`----`
	`1 Total to be deinstalled`

Step	Action
	+--+ Deinstalling Software... +--+ installp: DEINSTALLING software for: NetApp.snapdrive 2.1 Finished processing all filesets. (Total time: 1 secs). +--+ Summaries: +--+ Installation Summary -------------------- Name Level Part Event Result --- NetApp.snapdrive 2.1 USR DEINSTALL SUCCESS **Example of warning:** If you are trying to uninstall a product that is not currently installed, you see messages such as the following: Not Installed ------------- No software could be found on the system that could be deinstalled for the following requests: NetAppSnapdrive (The fileset may not be currently installed, or you may have made a typographical error.)

Installing and uninstalling SnapDrive for UNIX on an HP-UX host

System requirements for FCP/iSCSI configurations

The following table lists the minimum requirements for using SnapDrive with an HP-UX host in an FCP or iSCSI environment.

Component	Requirement
NetApp FCP HP-UX Attach Kit or iSCSI HP-UX Initiator Support Kit	To make sure you have the correct version of the kit, see the online *SnapDrive for UNIX Compatibility Matrix* and the *Compatibility and Configuration Guide for NetApp's FCP and iSCSI Products.* Set up the host and storage system according to the instructions in the documentation for your attach kit or support kit. You must do this before you install SnapDrive for UNIX.
Additional disk space	SnapDrive for UNIX maintains three log files. While SnapDrive for UNIX rotates the files when they reach a maximum size, you should make sure you have sufficient disk space for them. For more information about the log files, see "Setting up audit, recovery, and trace logging" on page 89. Based on the default settings for the audit and trace log files, you need at least 1.1 MB of space. There is no default size for the recovery log because it rotates only after an operation completes, not when it reaches a specific size.

Getting the software package

You can obtain the SnapDrive for UNIX software package in two ways:

- Download the package from http://now.netapp.com. See "Downloading the SnapDrive for UNIX software from NOW" on page 22.
- Get the package from the SnapDrive for UNIX CD-ROM provided with the media package for your host operating system. See "Getting SnapDrive for UNIX software from the CD-ROM" on page 23.

Moving the downloaded file to a local directory

If you downloaded the file and did not place it on your HP-UX machine, you must move it to that machine. Complete the following steps.

Step	Action

1	Use commands similar to the following ones to move the file you downloaded from the NOW site to your HP-UX machine.
	`# mkdir /tmp/hpux`
	`# cd /tmp/hpux`
	`# cp /u/hpux/HP-pkgs/NTAPsnapdrive_hpux_2_1.depot .`
	Make sure you include the period (.) at the end of the copy command line.

2	You're now ready to install SnapDrive for UNIX. Follow the steps in the section "Installing SnapDrive for UNIX on an HP-UX host" on page 41.

Installing SnapDrive for UNIX on an HP-UX host

To install SnapDrive for UNIX on an HP-UX host, complete the following steps.

Step	Action
1	Make sure you are logged in as either root or an authorized SD (software distributor) user for that HP-UX host. **Note** The installation file is a standard HP-UX depot file, so any user on your system with software installation privileges can perform the installation. For more information on authorized users, see the HP-UX `swacl` man page.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="3">2</td><td>If you are installing...</td><td>Then...</td></tr>
<tr><td>Software from the CD-ROM</td><td>
1. Mount the CD-ROM on the host system by entering a command similar to the following:
<pre># mount /dev/cdrom /mnt/cdrom</pre>
2. Change to the directory where your CD-ROM is mounted.
For example, change to /cdrom/cdrom0:
<pre>cd /cdrom/cdrom0</pre>
Note
You can install the software directly from the CD-ROM. NetApp, however, recommends that you copy the file to the local disk before installing.
3. If you want to install the software from a local directory on the host, copy the file to that directory.
For example, you might need to enter commands similar to the following:
<pre># mkdir /tmp/hpux
mount /dev/cdrom /mnt/cdrom
cp /mnt/cdrom/NTAPsnapdrive_hpux_2_1.depot
/tmp/hpux
cd /tmp/hpux</pre>
</td></tr>
<tr><td>Downloaded software</td><td>Change to the directory where you put the software you downloaded from the NOW site.</td></tr>
</table>

Step	Action

3	Enter the following command to install the software:
	`# swinstall -s <pathname>/NTAPsnapdrive_hpux_2_1.depot snapdrive`
	pathname is the full path to the NTAPsnapdrive file. This path must begin with a slash (/). The file needs to be on a NFS mounted directory. The script installs SnapDrive for UNIX in the /opt/NetApp/snapdrive/ directory.
	Example: The `swinstall` command installs the SnapDrive for UNIX software without a problem. It writes installation information to a log file.
	`# swinstall -s /tmp/hpux/NTAPsnapdrive_hpux_2_1.depot snapdrive`
	`=======  03/11/04 15:49:28 EST  BEGIN swinstall SESSION (non-interactive) (jobid=netapp2-0168)`
	`* Session started for user "root@netapp2".`
	`* Beginning Selection`
	`* Target connection succeeded for "netapp2:/".`
	`* Source: /tmp/hpux/NTAPsnapdrive_hpux_2_1.depot`
	`* Targets: netapp2:/`
	`* Software selections:`
	`snapdrive.command_parisc,r=1,fr=2.1,fa=HP-UX_B.11.00_32/64`
	`snapdrive.snapdrive_diag_scripts,r=1`
	`snapdrive.snapdrive_docs,r=1`
	`* Selection succeeded.`
	`* Beginning Analysis and Execution`
	`* Session selections have been saved in the file "/.sw/sessions/swinstall.last".`
	`* The analysis phase succeeded for "netapp2:/".`
	`* The execution phase succeeded for "netapp2:/".`
	`* Analysis and Execution succeeded.`
	`NOTE: More information may be found in the agent logfile using the command "swjob -a log netapp2-0168 @ netapp2:/".`
	`=======  03/11/04 15:49:48 EST  END swinstall SESSION (non-interactive) (jobid=netapp2-0168)`

Step	Action

4	Verify your installation using the `swverify` command.
	
	Example: The following example uses the `swverify` command to confirm that the SnapDrive for UNIX software installed.
	
	`# swverify snapdrive`
	
	`=======  03/12/04 11:55:49 EST  BEGIN swverify SESSION (non-interactive) (jobid=netappp2-0172)`
	
	`* Session started for user "root@netapp2".`
	
	`* Beginning Selection`
	`* Target connection succeeded for "netapp2:/".`
	`* Software selections:`
	`snapdrive.command_parisc,l=/opt/NetApp/snapdrive,r=2,fr=2.1,fa=HP-UX_B.11.00_32/64`
	`snapdrive.snapdrive_diag_scripts,l=/opt/NetApp/snapdrive,r=2.1`
	`snapdrive.snapdrive_docs,l=/opt/NetApp/snapdrive,r=1`
	`* Selection succeeded.`
	`* Beginning Analysis`
	`* Session selections have been saved in the file "/.sw/sessions/swverify.last".`
	`* The analysis phase succeeded for "netapp2:/".`
	`* Verification succeeded.`
	
	`NOTE:  More information may be found in the agent logfile using the command "swjob -a log netapp2-0172 @ netapp2:/".`
	
	`=======  03/12/04 11:55:54 EST  END swverify SESSION (non-interactive) (jobid=netapp2-0172)`
	
	`#`

Step	Action
5	Complete your setup by configuring SnapDrive for UNIX for your system. Most of this information is set by default; however, you need to specify the following information: ◆ The login information for the storage system. See "Specifying the current login information for storage systems" on page 111. ◆ The autosupport settings (autosupport is an optional feature, but NetApp recommends you enable it). See "Setting up autosupport" on page 97. **Note** For general information about configuration settings, see "Setting configuration information" on page 64.

Uninstalling the software from an HP-UX host

To uninstall SnapDrive for UNIX from an HP-UX system, complete the following steps.

Step	Action
1	Ensure that you are logged in as either root or an authorized SD (software distributor) user for that HP-UX host.
2	Use the `swremove` command to remove the NetApp software. `# swremove snapdrive` **Note** This command does not remove the log files. You must go to the /var/snapdrive directory and manually remove them.

Step	Action

	Example: The following example uninstalls SnapDrive for UNIX on an HP-UX system.
	
	`# swremove snapdrive`
	
	`=======  03/11/04 16:22:44 EST  BEGIN swremove SESSION (non-interactive) (jobid=netapp2-0170)`
	
	`* Session started for user "root@netapp2".`
	`* Beginning Selection`
	`* Target connection succeeded for "netapp2:/".`
	`* Software selections:`
	`snapdrive.command_parisc,l=/opt/NetApp/snapdrive,r=2,fr=2.1,fa=HP-UX_B.11.00_32/64`
	`snapdrive.snapdrive_diag_scripts,l=/opt/NetApp/snapdrive,r=1`
	`snapdrive.snapdrive_docs,l=/opt/NetApp/snapdrive,r=1`
	`* Selection succeeded.`
	`* Beginning Analysis`
	`* Session selections have been saved in the file "/.sw/sessions/swremove.last".`
	`* The analysis phase succeeded for "netapp2:/".`
	`* Analysis succeeded.`
	`* Beginning Execution`
	`* The execution phase succeeded for "netapp2:/".`
	`* Execution succeeded.`
	`NOTE: More information may be found in the agent logfile using the command "swjob -a log netapp2-0170 @ netapp2:/".`
	
	`=======  03/11/04 16:22:51 EST  END swremove SESSION (non-interactive) (jobid=netapp2-0170)`
	
	`#`

Installing and uninstalling SnapDrive for UNIX on a Linux host

System requirements for iSCSI configurations

The following table lists the minimum requirements for using SnapDrive with a Red Hat Enterprise Linux host in an iSCSI environment.

Component	Requirement
NetApp iSCSI Red Hat Linux Initiator Support Kit	To make sure you have the correct version of the kit, see the online *SnapDrive for UNIX Compatibility Matrix* and the *Compatibility and Configuration Guide for NetApp's FCP and iSCSI Products.* Set up the host and storage system according to the instructions in the installation and setup guide for the iSCSI Red Hat Linux kit. You must do this before you install SnapDrive for UNIX.
Additional disk space	SnapDrive for UNIX maintains three log files. While SnapDrive for UNIX rotates the files when they reach a maximum size, you should make sure you have sufficient disk space for them. For more information about the log files, see "Setting up audit, recovery, and trace logging" on page 89. Based on the default settings for the audit and trace log files, you need at least 1.1 MB of space. There is no default size for the recovery log because it rotates only after an operation completes, not when it reaches a specific size.

Getting the software package

You can obtain the SnapDrive for UNIX software package in two ways:

- Download the package from http://now.netapp.com. See "Downloading the SnapDrive for UNIX software from NOW" on page 22.
- Get the package from the SnapDrive for UNIX CD-ROM provided with the media package for your host operating system. See "Getting SnapDrive for UNIX software from the CD-ROM" on page 23.

Moving the downloaded file to a local directory

If you downloaded the file and did not place it on your Linux host, you must move it to that host. Complete the following steps.

Step	Action

1	Copy the downloaded file to your Linux host. You can place it in any directory on the host.
	
	Example: You can use commands similar to the following ones to move the file you downloaded from the NOW site to your Linux machine.
	
	`# mkdir /tmp/linux`
	`# cd /tmp/linux`
	`# cp /u/linux/netapp.snapdrive.linux_2_1.rpm .`
	
	Make sure you include the period (.) at the end of the copy command line.

2	You're now ready to install SnapDrive for UNIX. Follow the steps in the section "Installing SnapDrive for UNIX on an Linux host" on page 48.

Installing SnapDrive for UNIX on an Linux host

To install SnapDrive for UNIX on a Linux host, complete the following steps.

Step	Action
1	Make sure you are logged in as root. If you are executing this file remotely and your system configuration does not allow you to log in as root, use the `su` command to become root. **Note** The installation file is a standard Linux.rpm file.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="3">2</td><td>If you are installing...</td><td>Then...</td></tr>
<tr><td>Software from the CD-ROM</td><td>
1. Mount the CD-ROM on the host system by entering a command similar to the following:
<code># mount /dev/cdrom /mnt/cdrom</code>
2. Change to the directory where your CD-ROM is mounted.
For example, change to /cdrom/cdrom0:
<code>cd /cdrom/cdrom0</code>
Note
You can install the software directly from the CD-ROM or you can copy it to a local disk.
3. If you want to install the software from a local directory on the host, copy the file to that directory.
For example, you might need to enter commands similar to the following:
<code># mkdir /tmp/linux
mount /dev/cdrom /mnt/cdrom
cp /mnt/cdrom/netapp.snapdrive.linux_2_1.rpm /tmp/linux
cd /tmp/linux</code>
</td></tr>
<tr><td>Downloaded software</td><td>Change to the directory on your Linux host where you put the software you downloaded from the NOW site.</td></tr>
<tr><td>3</td><td colspan="2">Use the <code>rpm</code> command to install the software.
<code># rpm -U -v <<i>pathname</i>>/netapp.snapdrive.linux_2_1.rpm</code>
Example 1: The <code>rpm</code> command installs the SnapDrive for UNIX software without a problem. It writes installation information to a log file.
<code># rpm -U -v netapp.snapdrive.linux_2_1.rpm
    Preparing packages for installation...
    netapp.snapdrive-2.1
#</code></td></tr>
</table>

Step	Action

4	Verify your installation.
	
	Example: The following example uses the `rpm` and `ls` commands to verify that the SnapDrive for UNIX software installed.
	
	`# rpm -qa netapp.snapdrive`
	`netapp.snapdrive-2.1`
	`# ls -R /opt/NetApp`
	`/opt/NetApp:`
	`snapdrive`
	
	`/opt/NetApp/snapdrive:`
	`bin  diag  docs  snapdrive.conf`
	
	`/opt/NetApp/snapdrive/bin:`
	`snapdrive`
	
	`/opt/NetApp/snapdrive/diag:`
	`filer_info  linux_info  SHsupport.pm  snapdrive.dc  Telnet.pm`
	
	`/opt/NetApp/snapdrive/docs:`
	`man1`
	
	`/optNetApp/snapdrive/docs/man1:`
	`brocade_info.1  filer_info.1  linux_info.1  snapdrive.1  snapdrive.dc.1`
	`# ls -l /usr/sbin/snapdrive`
	`-rwxr-xr-x    1 root    root ../../opt/NetApp/snapdrive/bin/snapdrive`
	`#`

5	Complete your setup by configuring SnapDrive for UNIX for your system. Most of this information is set by default; however, you need to specify the following information: ◆ The login information for the storage system. See “Specifying the current login information for storage systems” on page 111. ◆ The autosupport settings (autosupport is an optional feature, but NetApp recommends you enable it). See “Setting up autosupport” on page 97. **Note** For general information about configuration settings, see “Setting configuration information” on page 64.

Uninstalling the software from a Linux host

To uninstall SnapDrive for UNIX from a Linux system, complete the following steps.

Step	Action
1	Ensure that you are logged in as root.

2	Use the `rpm` command to remove the NetApp software.
	
	Example: The following example uses the `rpm` command to uninstall the SnapDrive for UNIX software.
	
	`#  rpm -e netapp.snapdrive-2.1`
	`#`
	
	Note
	This command does not remove the log files. You must go to the /var/log directory and manually remove them.

3	Verify that the package uninstalled.
	
	Example: The following example verifies that SnapDrive for UNIX is no longer installed.
	
	`# rpm -qa netapp.snapdrive`
	`#`

Installing and uninstalling SnapDrive for UNIX on a Solaris host

System requirements for FCP configurations

The following table lists the minimum requirements for using SnapDrive with a Solaris host in an FCP environment.

Component	Requirement
NetApp FCP Solaris Attach Kit	To make sure you have the correct version of the kit, see the online *SnapDrive for UNIX Compatibility Matrix* and the *Compatibility and Configuration Guide for NetApp's FCP and iSCSI Products.* Set up the host and filer according to the instructions in the installation and setup guide in the Solaris kit. You must do this before you install SnapDrive for UNIX.
Additional disk space	SnapDrive for UNIX maintains three log files. While SnapDrive for UNIX rotates the files when they reach a maximum size, you should make sure you have sufficient disk space for them. For more information about the log files, see "Setting up audit, recovery, and trace logging" on page 89. Based on the default settings for the audit and trace log files, you need at least 1.1 MB of space. There is no default size for the recovery log because it rotates only after an operation completes, not when it reaches a specific size.

Getting the software package

You can obtain the SnapDrive for UNIX software package in two ways:

- Download the package from http://now.netapp.com. See "Downloading the SnapDrive for UNIX software from NOW" on page 22.
- Get the package from the SnapDrive for UNIX CD-ROM provided with the media package for your host operating system. See "Getting SnapDrive for UNIX software from the CD-ROM" on page 23.

Uncompressing the software package you downloaded

If you downloaded the software package from NOW, you can uncompress it by completing the following steps:

Step	Action
1	Change to the directory where you downloaded the compressed file. For example, change to the /tmp directory: `# cd /tmp`
2	Enter the following command to uncompress the file: `# uncompress NTAPsnapdrive.tar.Z`
3	Enter the following command to extract the software: `# tar -xvf NTAPsnapdrive.tar`
4	You're now ready to install SnapDrive for UNIX. Follow the steps in the section "Installing SnapDrive for UNIX on a Solaris host" on page 53.

Installing SnapDrive for UNIX on a Solaris host

To install SnapDrive for UNIX on a Solaris host, complete the following steps.

Step	Action	
1	Ensure that you are logged in as root.	
2	**If you are installing...**	**Then...**
	Software from the CD-ROM	**1.** Mount the CD-ROM on the host system. **2.** Change to the directory where your CD-ROM is mounted. For example, change to /cdrom/cdrom0: `cd /cdrom/cdrom0`
	Downloaded software from NOW	Change to the directory where you extracted the software.

Step	Action

3	Enter the following command to install the software:
	`# ./install`
	NetApp recommendsthat you have both the attach kit and the ASL installed before you install SnapDrive for UNIX.
	Example: The script installs the SnapDrive for UNIX software without a problem. It writes installation information to a log file.
	`# ./install`
	`Installing NTAPsnapdrive now ...`
	`NTAPsnapdrive install completed successfully.`
	`snapdrive Installation complete. Log is in`
	`/tmp/snapdrive_install_log.23752.`
	`#`

4	Complete your setup by configuring SnapDrive for UNIX for your system. Most of this information is set by default; however, you need to specify the following information: ◆ The login information for the storage system. See "Specifying the current login information for storage systems" on page 111. ◆ The autosupport settings (autosupport is an optional feature, but NetApp recommends you enable it). See "Setting up autosupport" on page 97. **Note** For general information about configuration settings, see "Setting configuration information" on page 64.

Uninstalling the software on a Solaris host

To uninstall SnapDrive for UNIX from a Solaris system, complete the following steps.

Step	Action
1	Ensure that you are logged in as root.

Step	Action

2	Use the `uninstall` command to remove the NetApp software. Answer `y` (or `yes`) when the command asks whether you are sure you want to remove the attach kit.
	
	If you answer `n` (or `no`), the command does not uninstall the attach kit.
	
	`# ./uninstall`
	`  Are you sure you want to uninstall the snapdrive package? [y,n,?,q] y`
	`  Removing NTAPsnapdrive ...`
	`  NTAPsnapdrive successfully removed.`
	`  Remove process complete. Log is in /tmp/snapdrive_remove_log.23789`
	`#`
	
	To avoid getting the confirmation query, enter the `./uninstall` command with the `-f` option, which prevents the uninstall script from running in interactive mode.
	
	Note
	The uninstall process might fail if the package has been partially uninstalled. If that happens, you can uninstall SnapDrive for UNIX by inserting the CD-ROM and executing the `./uninstall` command from the top-level directory.

Completing the installation

Completing the installation

Once you've installed SnapDrive for UNIX on your host, you need to complete the installation by performing the following tasks:

- Verify that the installation program installed all the necessary files on your host. For more information, see "Files installed by SnapDrive for UNIX on the host" on page 57.
- Confirm that the configuration variables in the snapdrive.conf have the correct settings. For the majority of these variables, the default values should be correct. In a few cases, you may need to modify these values. See "Setting configuration information" on page 64.
- Supply SnapDrive for UNIX with the current storage system login information. When you set up your storage system, you supplied a user login for it. SnapDrive for UNIX needs this login information in order to work with the storage system,. For information on supplying the login information, see "Specifying the current login information for storage systems" on page 111.
- Specify the access control permissions for each host on each storage system. If you do not specify access control permissions, SnapDrive for UNIX checks the value of the variable `all-access-if-rbac-unspecified` in the snapdrive.conf file. If the variable is set to on (the default), it allows the host to perform all SnapDrive for UNIX operations on that storage system. If the value is set to off, it prevents the host from performing any operations other than the show/list operations. For information on setting permissions, see "Setting up access control" on page 105.

Files installed by SnapDrive for UNIX on the host

Files installed by the software

When you install SnapDrive for UNIX, it installs a number of files on the host. These files serve different purposes.

Note

There are slight differences in the files based on your host operating system. In most cases, these differences have to do with the path names used by that host. Also, some hosts use symbolic links (symlinks) to files while other hosts simply install the file in two locations.

Executables: These are the executables SnapDrive for UNIX uses.

AIX

- /opt/NetApp/snapdrive/bin/snapdrive

HP-UX

- /opt/NetApp/snapdrive/bin/snapdrive
- /opt/NetApp/snapdrive/bin/snapdrive_parisc

Linux

- /opt/NetApp/snapdrive/bin/snapdrive

Solaris

- /opt/NTAPsnapdrive/bin/snapdrive

Configuration file: SnapDrive for UNIX stores configuration information in this file. You should modify it for your system. If you upgrade your version of SnapDrive for UNIX, it maintains your current snapdrive.conf file. For more information on setting configuration variables see "Setting configuration information" on page 64. For information on upgrading your version of SnapDrive for UNIX and how it handles the snapdrive.conf file, see "Upgrading to a new version of SnapDrive for UNIX" on page 61.

AIX, HP-UX, and Linux

- /opt/NetApp/snapdrive/snapdrive.conf

Solaris

- /opt/NTAP/snapdrive/snapdrive.conf

Uninstall files: If you decide to remove SnapDrive for UNIX, it uses these files. These files are also in the top-level directory on the SnapDrive for UNIX CD-ROM. See "Installing SnapDrive for UNIX on a Solaris host" on page 53 for more information.

AIX

You use SMIT to uninstall SnapDrive for UNIX from an AIX host so there is not an uninstall file.

HP-UX

You use swremove to uninstall SnapDrive for UNIX from a HP-UX host so there is not an uninstall file.

Linux

You use rpm to uninstall SnapDrive for UNIX from a Linux host so there is not an uninstall file.

Solaris

- /opt/NTAP/snapdrive/bin/snapdrive.admin
- /opt/NTAP/snapdrive/bin/uninstall

Diagnostic files: You can run the following files if you have a problem with SnapDrive. For more information about these files, see "Data collection utility" on page 220.

AIX and HP-UX

- /opt/NetApp/snapdrive/diag/snapdrive.dc
- /opt/NetApp/snapdrive/diag/filer_info
- /opt/NetApp/snapdrive/diag/brocade_info
- /opt/NetApp/snapdrive/diag/cisco_info
- /opt/NetApp/snapdrive/diag/mcdata_info
- /opt/NetApp/snapdrive/diag/SHsupport.pm
- /opt/NetApp/snapdrive/diag/Telnet.pm

AIX only

- /opt/NetApp/snapdrive/diag/aix_info

HP-UX only

- /opt/NetApp/snapdrive/diag/hpux_info

Linux

- /opt/NetApp/snapdrive/diag/snapdrive.dc
- /opt/NetApp/snapdrive/diag/filer_info
- /opt/NetApp/snapdrive/diag/linux_info
- /opt/NetApp/snapdrive/diag/SHsupport.pm
- /opt/NetApp/snapdrive/diag/Telnet.pm

Solaris

- /opt/NTAPsnapdrive/diag/snapdrive.dc
- /opt/NTAPsnapdrive/diag/solaris_info
- /opt/NTAPsnapdrive/diag/filer_info
- /opt/NTAPsnapdrive/diag/brocade_info
- /opt/NTAPsnapdrive/diag/cisco_info
- /opt/NTAPsnapdrive/diag/mcdata_info
- /opt/NTAPsnapdrive/diag/SHsupport.pm
- /opt/NTAPsnapdrive/diag/Telnet.pm

Log files: SnapDrive for UNIX creates several log files.

AIX and HP-UX

- /var/snapdrive/sd-audit.log
- /var/snapdrive/sd-recovery.log
- /var/snapdrive/sd-trace.log

Log files: SnapDrive for UNIX creates several log files.

AIX and HP-UX

- /var/snapdrive/sd-audit.log
- /var/snapdrive/sd-recovery.log
- /var/snapdrive/sd-trace.log

Linux and Solaris

- /var/log/sd-audit.log
- /var/log/sd-recovery.log
- /var/log/sd-trace.log

Man pages: SnapDrive for UNIX provides man pages in several formats:

AIX and HP-UX

- /opt/NetApp/snapdrive/man/man1/snapdrive.dc.1

- /opt/NetApp/snapdrive/man/man1/snapdrive.1
- /opt/NetApp/snapdrive/docs/snapdrive.1.cat
- /opt/NetApp/snapdrive/docs/snapdrive.1.html
- /opt/NetApp/snapdrive/docs/snapdrive.1.pdf
- /opt/NetApp/snapdrive/docs/snapdrive.1.txt
- /opt/NetApp/snapdrive/docs/brocade_info.1
- /opt/NetApp/snapdrive/docs/mcdata_info.1
- /opt/NetApp/snapdrive/docs/cisco_info.1
- /opt/NetApp/snapdrive/docs/filer_info.1
- /opt/NetApp/snapdrive/docs/aix_info.1 (AIX host only)
- /opt/NetApp/snapdrive/docs/hpux_info.1 (HP-UX host only)

Linux

- /opt/NetApp/snapdrive/docs/man1/snapdrive.dc.1
- /opt/NetApp/snapdrive/docs/man1/snapdrive.1
- /opt/NetApp/snapdrive/docs/man1/filer_info.1
- /opt/NetApp/snapdrive/docs/man1/linux_info.1

Solaris

- /usr/share/man/man1/snapdrive.1
- /opt/NTAPsnapdrive/docs/snapdrive.1.man
- /opt/NTAPsnapdrive/docs/snapdrive.1.cat
- /opt/NTAPsnapdrive/docs/snapdrive.1.txt
- /opt/NTAPsnapdrive/docs/snapdrive.1.pdf
- /opt/NTAPsnapdrive/docs/snapdrive.1.html
- /opt/NTAPsnapdrive/docs/solaris_info.1
- /opt/NTAPsnapdrive/docs/brocade_info.1
- /opt/NTAPsnapdrive/docs/mcdata_info.1
- /opt/NTAPsnapdrive/docs/cisco_info.1
- /opt/NTAPsnapdrive/docs/filer_info.1

Upgrading to a new version of SnapDrive for UNIX

Upgrading your version of SnapDrive for UNIX

SnapDrive for UNIX makes it easy for you to upgrade from an earlier version, such as 1.1, to the current version.

You do not have to uninstall SnapDrive for UNIX. Instead, install the latest version of the software on top of the current version.

When you install a new version, SnapDrive for UNIX checks to see if you already have a version installed. If you do, it preserves the current snapdrive.conf file and renames the version of the file it is installing to snapdrive.conf.2.1. This way it avoids overwriting your existing snapdrive.conf file, so you do not lose any changes you made if you customized your settings in the file.

By default, SnapDrive for UNIX comments out the variables in the snapdrive.conf file. This means it automatically uses the default values for all variables except the ones you customize. As a result, SnapDrive for UNIX will use the default values for variables that are new even if you have an earlier version of the snapdrive.conf file. If you want to change these values, you must add the variables to your current snapdrive.conf file and specify the values you want.

For more information on the snapdrive.conf file and its variables, see "Setting configuration information" on page 64.

Changed variables in the snapdrive.conf file

In the previous release, SnapDrive for UNIX automatically applied space reservations when you used the `snapdrive storage create` command to provision storage, and the `snapdrive snap connect` command to connect a snapshot to a new location. Customers who did not wish to enable space reservations for these two commands could turn them off using the `space-reservations-enabled=on |off` variable in the snapdrive.conf file.

The `space-reservations-enabled=on |off` is supported in the current version of SnapDrive for UNIX for backward compatibility, but will be removed in a subsequent release. NetApp recommends that you use two new command-line options,`-reserve` and `-noreserve` to enable or disable LUN space reservations in SnapDrive `snapdrive storage create and snap connect` commands. By default, SnapDrive for UNIX sets space reservations to "on" when you enter the `snapdrive storage create` command, and to "off" when you enter `snapdrive snap connect`.

Configuring and Using SnapDrive for UNIX 3

About this chapter

This chapter provides details about setting your SnapDrive for UNIX configuration options and general information about using SnapDrive for UNIX.

Topics in this chapter

This chapter discusses the following topics:

- "Setting configuration information" on page 64
- "Preparing hosts for adding LUNs" on page 88
- "Setting up audit, recovery, and trace logging" on page 89
- "Setting up autosupport" on page 97
- "General steps for executing commands" on page 100

Setting configuration information

Using the snapdrive.conf file

The snapdrive.conf file controls the configurable variables available in SnapDrive for UNIX. This file contains a name/value pair for each configurable variable.

You use a text editor to modify this file. SnapDrive for UNIX automatically checks the information in this file each time it starts.

The snapdrive.conf file is in the SnapDrive for UNIX installation directory (see the installation instructions for your operating system for the complete path to this directory).

SnapDrive for UNIX also provides some commands you can use to work with this file, such as the `snapdrive config show` command, which displays this file.

Verify the settings in snapdrive.conf

The snapdrive.conf file comes with most of the variables set to default values. In most cases, these are the values you should use when you run SnapDrive for UNIX. To complete your SnapDrive for UNIX installation, though, you should verify the default values in the snapdrive.conf file. In particular, you should check the values for the following variables:

- Logging
- Autosupport
- Accurate path information for the system

To view the current settings, execute the `snapdrive config show` command.

Some operations available with snapdrive config command

The `snapdrive config` command has several formats. These commands allow you to perform a number of configuration tasks, including working with the snapdrive.conf file, setting the user/password information SnapDrive for UNIX needs to access the storage systems, setting access control permissions, and determining LUN configuration information for your system. The following is a brief overview of the `snapdrive config` command formats:

- `snapdrive config show` [*host_file_name*]

  This command displays the snapdrive.conf current name/value pairs and their defaults. If you include the *host_file_name* argument, it writes the information to the file you specify on the host.

- `snapdrive config list`

  `snapdrive config set` *user_name filername* [*filername* . . .]

  `snapdrive config delete` *filername* [*filername* . . .]

  These versions of the `config` command handle user logins for the storage systems.

  You can use the `snapdrive config list` command to display the user names that have been configured in SnapDrive for UNIX for accessing storage systems.

  The `set` version of the command lets you tell SnapDrive for UNIX which user name/password pair to use to access a storage system while the `delete` version removes the specified user name/password pair from SnapDrive for UNIX.

  For more information, see "Specifying the current login information for storage systems" on page 111.

- `snapdrive config access {list | show}` *filername*

  This operation lets you display the access control settings for a host trying to access a storage system.

  For more information, see "Setting up access control" on page 105.

- `snapdrive config prepare luns -count` *count*

  `snapdrive config check luns`

  These commands let you prepare the host for creating a specific number of LUNs as well as to determine how many LUNs you could create.

  For more information, see "Preparing hosts for adding LUNs" on page 88.

  Note

  You only need these commands on Linux and Solaris hosts. AIX and HP-UX hosts do not require any preparation prior to creating LUNs.

Determining options and their default values

The supported configurable items and their default settings can vary across host operating systems and the different versions of SnapDrive for UNIX. An easy way to determine current configurable items and their settings is to execute the

`snapdrive config show` command. The following example shows output that the `config show` option can produce. You should run the `snapdrive config show` command to get a current copy of the file for your host platform.

Note

If you are running SnapDrive for UNIX on a different host operating system, some of the defaults may be different. For example, on an AIX or HP-UX host, the default path for a log file is /var/snapdrive/... instead of the Linux or Solaris default path of /var/log/... .

```
#
# Snapdrive Configuration
# file: /opt/NetApp/snapdrive/snapdrive.conf
# Version 2.1 (Change 460739 Built Wed Oct 5 16:05:31 EDT 2005)
#
#
# Default values are shown by lines which are commented-out in this file.
# If there is no un-commented-out line in this file relating to a particular value, then
# the default value represented in the commented-out line is what SnapDrive will use.
#
# To change a value:
#
# -- copy the line that is commented out to another line
# -- Leave the commented-out line
# -- Modify the new line to remove the '#' and to set the new value.
# -- Save the file and exit
#
audit-log-file="/var/log/sd-audit.log" # audit log file
trace-log-file="/var/log/sd-trace.log" # trace log file
recovery-log-file="/var/log/sd-recovery.log" # recovery log file
autosupport-enabled=off # Enable autosupport (requires autosupport-filer be set)
autosupport-filer="" # Filer to use for autosupport (filer must be configured for
autosupport)
audit-log-max-size=20480 # Maximum size (in bytes) of audit log file
audit-log-save=2 # Number of old copies of audit log file to save
available-lun-reserve=8 # Number of LUNs for which to reserve host resources
device-retries=3 # Number of retries on Ontap filer LUN device inquiry
device-retry-sleep-secs=1 # Number of seconds between Ontap filer LUN device inquiry
retries
enable-implicit-host-preparation=off # Enable implicit host preparation for LUN
creation
filer-restore-retries=140 # Number of retries while doing lun restore
filer-restore-retry-sleep-secs=15 # Number of secs between retries while restoring lun
filesystem-freeze-timeout-secs=300 # File system freeze timeout in seconds
```

```
default-noprompt=off # A default value for -noprompt option in the command line
mgmt-retries=3 # Number of retries on ManageONTAP control channel
mgmt-retry-sleep-secs=2 # Number of seconds between retries on ManageONTAP control
channel
mgmt-retry-sleep-long-secs=90 # Number of seconds between retries on ManageONTAP
control channel (failover error)
prepare-lun-count=16 # Number of LUNs for which to request host preparation
PATH="/sbin:/usr/sbin:/bin:/usr/bin:/opt/sanlun/bin" # toolset search path
password-file="/opt/NetApp/snapdrive/.pwfile" # location of password file
prefix-filer-lun="" # Prefix for all filer LUN names internally generated by storage
create
recovery-log-save=20 # Number of old copies of recovery log file to save
snapcreate-consistency-retries=3 # Number of retries on snapshot consistency check
failure
snapdelete-delete-rollback-with-snap=off # Delete all rollback snapshots related to
specified snapshot
snapmirror-dest-multiple-filervolumes-enabled=off # Enable snap restore and snap
connect commands to deal with snapshots moved to another filer volume (e.g. via
SnapMirror) where snapshot spans multiple filers or volumes.
snaprestore-snapmirror-check=on # Enable snapmirror destination volume check in snap
restore
snaprestore-delete-rollback-after-restore=on # Delete rollback snapshot after a
successfull restore
snaprestore-make-rollback=on # Create snap rollback before restore
snaprestore-must-make-rollback=on # Do not continue 'snap restore' if rollback creation
fails
space-reservations-enabled=on # Enable space reservations when creating new luns
default-transport="iscsi" # Transport type to use for storage provisioning, when a
decision is needed
multipathing-type="none" # Multipathing software to use when more than one multipathing
solution is available
trace-enabled=on # Enable trace
trace-level=5 # Trace levels: 1=FatalError; 2=AdminError; 3=CommandError; 4=warning,
5=info, 6=verbose
trace-log-max-size=0 # Maximum size of trace log file in bytes; 0 means one trace log
file per command
trace-log-save=20 # Number of old copies of trace log file to save
all-access-if-rbac-unspecified=on # Allow all access if the RBAC permissions file is
missing
use-https-to-filer=off # Communication with filer done via HTTPS instead of HTTP
```

The following table describes the configuration variables in this file.

Option	Description
all-access-if-rbac-unspecified=on	Set this value to either On or Off where ◆ On tells SnapDrive for UNIX to enable all access permissions if there is no access control permissions file on the storage system. This is the default value. ◆ Off specifies that the storage system allow the host only the permissions specified in the access control permissions file. You can specify access control permissions for each host where SnapDrive for UNIX runs by entering the permission string in an access control file. The string you specify controls which SnapDrive for UNIX snapshot and storage operations that host may perform on a given storage system. (These access permissions do not affect the show/list operations.) If you provide an access control file, this option has no effect. For more information on access control and how to set it up, see “Setting up access control” on page 105.
audit-log-file="/var/log/sd-audit.log"	Specify the location where SnapDrive for UNIX writes the audit log file. The default value depends on your host operating system. The path shown in this example is the default path for a Linux or Solaris host. For an AIX or HP-UX host, the default path is /var/snapdrive/sd-audit.log. See “Changing the defaults for audit logs” on page 92 for more information.

Option	Description
audit-log-max-size=20480	Specify the maximum size, in bytes, of the audit log file. When the file reaches this size, SnapDrive for UNIX renames it and starts a new audit log. The default value is 20,480 bytes. Because SnapDrive for UNIX never starts a new log file in the middle of an operation, the actual size of the file could vary slightly from the value specified here. **Note** NetApp recommends that you use the default value. If you decide to change the default, keep in mind that having a large number of large log files can take up space on your disk and might eventually affect performance.
audit-log-save=2	Specify how many old audit log files SnapDrive for UNIX should save. After this limit is reached, SnapDrive for UNIX discards the oldest file when it creates a new one. SnapDrive for UNIX rotates this file based on the value you specify for the option audit-log-max-size. (For more information on log rotation, see "Settings affecting log file rotation" on page 91.) The default value is 2. **Note** NetApp recommends that you use the default value. If you decide to change the default, keep in mind that having a large number of large log files can take up space on your disk and might eventually affect performance.

Option	Description
autosupport-enabled=off	Set this value to either On to enable autosupport or Off to disable it. By default this option is off because you need to perform additional setup steps to enable it, including specifying a value for the autosupport-filer option. NetApp recommends that you enable this option. See “Setting up autosupport” on page 97 for details about how to enable this option.
autosupport-filer=" "	Enter the name or IP address of the storage system that autosupport should use to send the message. To disable this option, leave the storage system name blank. This option is disabled by default because it requires information specific to your setup. NetApp recommends that you enable this option. To enable autosupport, you must enter a value here and also set autosupport-enabled to On. See “Setting up autosupport” on page 97 for details about how to enable this option.

Option	Description
available-lun-reserve=8	Enter the number of LUNs that the host must be prepared to create when the current SnapDrive for UNIX operation completes. If too few operating system resources are available to create the number of LUNs specified, SnapDrive for UNIX requests additional resources, based on the value supplied with the enable-implicit-host-preparation variable. The default value is 8. **Note** This variable applies only to systems where host preparation is needed before you can create LUNs. Currently, only Linux and Solaris hosts require that preparation. This variable is only used on configurations that include LUNs.
default-noprompt=Off	Specify whether you want the `-noprompt` option to be available or not. By default, this option is not available (Off). If you turn this option On, then SnapDrive for UNXI does not prompt you to confirm an action requested by `-force`.

Option	Description
default-transport="iscsi"	Specify which protocol SnapDrive for UNIX uses as the transport type when creating storage, if a decision is required. The acceptable values are iscsi or fcp. The default is iscsi. **Note** On AIX hosts, you must also make sure the multipathing-type option is set correctly. If you specify “fcp” here, you must set multipathing-type to “SANPath”. If you specify “iscsi” here, you must specify “none” as the value for multipathing-type.
device-retries=3	Specify the number of times SnapDrive for UNIX attempts to inquire about the device where the LUN is. The default value is 3. In normal circumstances, the default value should be adequate. Under other circumstances, LUN queries for a `snap create` operation could fail simply because the storage system is exceptionally busy. If the LUN queries keep failing even though the LUNs are online and correctly configured, you might want to increase the number of retries. This variable is only used on configurations that include LUNs.

Option	Description
device-retry-sleep-secs=1	Specify the number of seconds SnapDrive for UNIX waits between inquiries about the device where the LUN resides. The default value is 1 second. In normal circumstances, the default value should be adequate. Under other circumstances, LUN queries for a `snap create` operation could fail simply because the storage system is exceptionally busy. If the LUN queries keep failing even though the LUNs are online and correctly configured, you might want to increase the number of seconds between retries. This variable is only used on configurations that include LUNs.

Option	Description
enable-implicit-host-preparation="on"	Specify whether SnapDrive for UNIX implicitly requests host preparation for LUNs or simply notifies you that it is required and exits. When this value is set to On, SnapDrive for UNIX implicitly requests the host to make more resources available if there will not be enough resources to create the correct number of LUNs once the current command completes. The number of LUNs being created is specified in the available-lun-reserve variable. When this variable is set to Off, SnapDrive for UNIX informs you if additional host preparation is necessary for the LUN creation and exits. You can then perform the operations necessary to free up resources needed for the LUN creation. For example, you can execute the `snapdrive config prepare luns` command. Once the preparation is complete, you can re-enter the current SnapDrive for UNIX command. By default, this variable is set to On. **Note** This variable applies only to systems where host preparation is needed before you can create LUNs. Currently, only Linux and Solaris hosts require that preparation. This variable is only used on configurations that include LUNs.

Option	Description
filer-restore-retries=140	Specify the number of times SnapDrive for UNIX attempts to restore a snapshot on a storage system if a failure occurs during the restore. The default value is 140. In normal circumstances, the default value should be adequate. Under other circumstances, this operation could fail simply because the storage system is exceptionally busy. If it keeps failing even though the LUNs are online and correctly configured, you might want to increase the number of retries.
filer-restore-retry-sleep-secs=15	Specify the number of seconds SnapDrive for UNIX waits between attempts to restore a snapshot. The default value is 15 seconds. In normal circumstances, the default value should be adequate. Under other circumstances, this operation could fail simply because the storage system is exceptionally busy. If it keeps failing even though the LUNs are online and correctly configured, you might want to increase the number of seconds between retries.
filesystem-freeze-timeout-secs=300	Specify the amount of time, in seconds, that SnapDrive for UNIX should wait when it can't access the file system before trying again.The default value is 300 seconds (5 minutes). This variable is only used on configurations that include LUNs.
mgmt-retries=3	Specify the number of times SnapDrive for UNIX retries an operation on the ManageONTAP control channel after a failure occurs. The default value is 3.

Option	Description
mgmt-retry-sleep-secs=2	Specify the number of seconds SnapDrive for UNIX waits before retrying an operation on the ManageONTAP control channel. The default value is 2 seconds.
mgmt-retry-sleep-long-secs=90	Specify the number of seconds SnapDrive for UNIX waits before retrying an operation on the ManageONTAP control channel after a failover error occurs. The default value is 90 seconds.
multipathing-type=" "	Specify the multipathing software to use. This option applies only if there is more than one multipathing solution available. If a platform supports only one type of multipathing, it does not use this option. For this release, the AIX platform is the only one where you must set this value. Solaris and HP-UX platforms provide support for only one type of multipathing each. The Linux platform does not support multipathing, so the default value there is “none”.The value you set for AIX depends on which protocol you are using. If you are using AIX with ◆ FCP, set this value to “SANPath”. In addition, set the default-transport option to “fcp”. ◆ iSCSI, set this value to “none”. In addition set the default-transport option to “iscsi”. **Note** For this release, multipathing is not supported on AIX hosts running iSCSI. This variable is only used on configurations that include LUNs.

Option	Description
path="/sbin:/usr/sbin:/bin:/usr/lib/ vxvm/diag.d:/usr/bin:/usr/local/bin"	Specify the search path the system uses to look for tools. NetApp recommends that you verify that this is the correct path for your system. If it is incorrect, change it to the correct path. The default value may vary depending on your operating system. In this case, this is the default path for a Solaris system or an HP-UX system. AIX systems don't use this variable because they process the commands differently.
password-file="/opt/NTAPsnapdrive/.pwfile"	Specify the location of password file for the user logins for the storage systems. The default value depends on your operating system. For example, this is the default path for a Solaris system. Some hosts use a default path of /opt/NetApp/snapdrive/.pwfile.
prefix-filer-lun=" "	Specify the prefix that SnapDrive for UNIX applies to all LUN names it generates internally. The default value for this prefix is the empty string. This variable allows the names of all LUNs created from the current host, but not explicitly named on a SnapDrive for UNIX command line, to share an initial string. This variable is only used on configurations that include LUNs.

Option	Description
prepare-lun-count=16	Specify how many LUNs SnapDrive for UNIX should prepare to create. SnapDrive for UNIX checks this value when there is a request to prepare the host to create additional LUNs. By default, this variable is set to 16. That means the system will be able to create 16 additional LUNs once the preparation is complete. **Note** This variable applies only to systems where host preparation is needed before you can create LUNs. Currently, only Linux and Solaris hosts require that preparation. This variable is only used on configurations that include LUNs.
recovery-log-file="/var/log/sd-recovery.log"	Specify the location where SnapDrive for UNIX writes the recovery log file. The default value depends on your host operating system. The path shown in this example is the default path for a Linux or Solaris host. For an AIX or HP-UX host, the default path is /var/snapdrive/sd-recovery.log. See “Contents of a recovery log” on page 93 for more information.

Option	Description
recovery-log-save=20	Specify how many old recovery log files SnapDrive for UNIX should save. After this limit is reached, SnapDrive for UNIX discards the oldest file when it creates a new one. SnapDrive for UNIX rotates this log file each time it starts a new operation. (For more information on log rotation, see "Settings affecting log file rotation" on page 91.) The default value is 20. **Note** NetApp recommends that you use the default value. If you decide to change the default, keep in mind that having a large number of large log files can take up space on your disk and might eventually affect performance.
snapcreate-must-make-snapinfo-on-qtree=off	Set this value to On to enable the `snapdrive snap create` operation to create snapshot information on a qtree. Set it to Off to disable this feature. This option is disabled by default. SnapDrive for UNIX always attempts to write snapinfo at the root of a qtree if the LUNs being snapped are at the root of that qtree. Setting this option to On means that SnapDrive for UNIX fails the `snap create` operation if it can't write this data. You should only turn this option on if you are mirroring snapshots using Qtree SnapMirror® (QSM) software. **Note** Snapshots of qtrees work the same way snapshots of volumes do.

Option	Description
snapcreate-consistency-retries=3	Specify the number of times SnapDrive for UNIX attempts a consistency check on a snapshot after it receives a message that a consistency check failed. This option is particularly useful on host platforms that do not include a freeze function. Currently, Linux is the only platform without that function. This variable is only used on configurations that include LUNs. For more information, see “Crash-consistent snapshots” on page 118. The default value for this option is 3.
snapdelete-delete-rollback-with-snap=off	Set this value to On to delete all rollback snapshots related to a specified snapshot. Set it to Off to disable this feature. This option is disabled by default. This option only takes effect during a `snap delete` operation. is used by the recovery log file if you encounter a problem with an operation. NetApp recommends that you accept the default setting.
snapmirror-dest-multiple-filervolumes-enabled=off	Set this value to On to restore snapshots that span multiple storage systems or volumes on (mirrored) destination storage systems.

Option	Description
snaprestore-delete-rollback-after-restore=on	Set this value to On to delete all rollback snapshots after a successful `snap restore` operation. Set it to Off to disable this feature. This option is enabled by default. This option is used by the recovery log file if you encounter a problem with an operation. NetApp recommends that you accept the default setting.
snaprestore-make-rollback=on	Set this value to either On to create a rollback snapshot or Off to disable this feature. This option is enabled by default. A rollback is a copy of the data on the storage system that SnapDrive for UNIX makes before it begins a `snap restore` operation. That way, if a problem occurs during the `snap restore` operation, you can use the rollback snapshot to restore the data to the state it was in before the operation began. If you do not want the extra security of a rollback snapshot at restore time, set this option to Off. If you like the rollback, but not enough to fail your `snap restore` operation if you can't make one, set the option snaprestore-must-make-rollback to Off. This option is used by the recovery log file, which you send to technical support if you encounter a problem. See "Contents of a recovery log" on page 93. NetApp recommends that you accept the default setting.

Option	Description
snaprestore-must-make-rollback=on	Set this value to On to fail a `snap restore` operation if the rollback creation fails or Off to disable this feature. The option is enabled by default. When you set this option to On, SnapDrive for UNIX attempts to make a rollback copy of the data on the storage system before it begins the `snap restore` operation. If it can't make a rollback copy of the data, SnapDrive for UNIX halts the `snap restore` operation. If you want the extra security of a rollback snapshot at restore time, but not enough to fail your `snap restore` operation if you can't make one, set this option to Off. This option is used by the recovery log file if you encounter a problem with an operation. NetApp recommends that you accept the default setting.
snaprestore-snapmirror-check=on	Set this value to on to enable the `snap restore` command to check the SnapMirror destination volume. If it is Off, the `snap restore` command is unable to check the destination volume. The default value is On.

Option	Description
space-reservations-enabled=on	This parameter was used to disable the space reservation for LUNs created by the `snapdrive snap connect` command and `snapdrive storage create` command. This parameter is supported in the current version of SnapDrive for UNIX for backward compatibility, but will be removed in a subsequent release. NetApp recommends using two new options, `-reserve` and `-noreserve` to enable or disable LUN space reservations in SnapDrive commands.By default, SnapDrive for UNIX sets space reservations to "on" when you enter the `snapdrive storage create` command, and to "off" when you enter `snapdrive snap connect`. You should do not need to adjust this parameter.
trace-enabled=on	Set this value to On to enable the trace log file or Off to disable it. This file is used by NetApp. The default value is On. Enabling this file does not affect performance. For more information, see "About the trace log file" on page 95.
trace-level=5	Specifies the types of messages SnapDrive for UNIX writes to the trace log file. This option accepts the following values: ◆ 1—Record fatal errors. ◆ 2—Record admin errors. ◆ 3—Record command errors. ◆ 4—Record warnings. ◆ 5—Record information messages. ◆ 6—Record in verbose mode. The default value is 5.

Option	Description
trace-log-file="/var/log/sd-trace.log"	Specify the location where SnapDrive for UNIX writes the trace log file. The default value depends on your host operating system. The path shown in this example is the default path for a Linux or Solaris host. For an AIX or HP-UX host, the default path is /var/snapdrive/sd-trace.log. See “About the trace log file” on page 95 for more information.
trace-log-max-size=0	Specify the maximum size of the trace log file in bytes. When the file reaches this size, SnapDrive for UNIX renames it and starts a new trace log. The default value is zero. This value means that, for every command, SnapDrive for UNIX creates a separate trace file. Because SnapDrive for UNIX never starts a new log file in the middle of an operation, the actual size of the file could vary slightly from the value specified here. **Note** NetApp recommends that you use the default value. If you decide to change the default, keep in mind that having a large number of large log files can take up space on your disk and might eventually affect performance.

Option	Description
trace-log-save=20	Specify how many old trace log files SnapDrive for UNIX should save. After this limit is reached, SnapDrive for UNIX discards the oldest file when it creates a new one. This option works with trace-log-max-size. By default, trace-log-max-size=0 saves one command in each file, and trace-log-save=20 retains the last log 20 files. For more information on log file, see "Settings affecting log file rotation" on page 91.
use-https-to-filer=off	Specify whether you want SnapDrive for UNIX to use SSL encryption (HTTPS) when it communicates with the storage system. By default, this value is set to off. **Note** If you are using a version of Data ONTAP prior to 7.0, you might see slower performance with HTTPS enabled. This is not an issue if you are running Data ONTAP 7.0. For more information on the security features available with SnapDrive for UNIX, see the chapter "Setting Up Security Features" on page 103.

Setting values in snapdrive.conf

To change the values in the snapdrive.conf file or add new name/value pairs, complete the following steps.

Step	Action
1	Log on as root.
2	Make a backup of the snapdrive.conf file.

Step	Action
3	Open the snapdrive.conf file in a text editor.
4	Make the changes to the file. To add a name/value pair, use the following format: *config-option-name*=*value* # optional comment *config-option-name* is the name of the option you want to configure; for example, audit-log-file. *value* is the value you want to assign to this option. If you want to include a comment with the name/value pair, precede the comment with a pound sign (#). You should enter only one name/value pair per line. If the name or the value uses a string, enclose the string in either single (') or double (") quotes. You can place the quotes around either the entire name/value pair or just the value. The following are three examples of how you can use quotes and comments with name/value pairs: "config-option-one=string with white space" # double quotes around the pair config-option-two="string with white space" # double quotes around the value config-option-2B='string with white space' # single quotes around the value
5	To modify an existing name/value pair, replace the current value with the new value. The best way to do this is to perform the following steps; **1.** Comment out the line you want to modify. **2.** Copy the commented-out line. **3.** Un-comment the copy by removing the pound sign. **4.** Modify the value. This way you always have a record of the default value in the file. If you want to specify a blank value (for example, you want to disable the audit log file), enter a pair of quotation marks ("").
6	Save the file after you make your changes. SnapDrive for UNIX automatically checks this file each time it start. Your changes take effect the next time it starts.

Checking your version of SnapDrive

To determine which version of SnapDrive you're using, complete the following step.

Step	Action

1	Enter the following command:
	`snapdrive version`
	Example: SnapDrive for UNIX displays its version information when you enter this command.
	`# snapdrive version`
	`  Version 2.0`
	`#`
	Note
	The only argument this command accepts is `-v` (verbose), which displays additional version details. If you include additional arguments, SnapDrive for UNIX displays a warning and then the version number.

Preparing hosts for adding LUNs

Checking host information

If your operating system requires that you prepare it before you create new LUNs, you can use the `snapdrive config` command. This command lets you check information about how many LUNs can be created on a storage system that is mapped to your host.

Not all host platform operating systems require that you prepare the host. Currently, these commands are required only on Linux and Solaris hosts.

Determine how many LUNs could be created

SnapDrive for UNIX lets you determine how many LUNs could be created on the host without exceeding a host-local limit. Use the following command to determine this value:

```
snapdrive config check luns
```

On a Linux host, this command checks the existing /dev/sg files to determine how many are unused.

On a Solaris host, this command scans /kernel/drv/sd.conf to determine how many unused entries it has that would be suitable for LUNs.

Add host entries for new LUNs

You can also make sure the host is prepared for the creation of a specific number of new LUNs. These LUNs will reside on a storage system that is currently mapped to the host. Use the following command:

```
snapdrive config prepare luns -count count
```

count is the number of new LUNs for which you want the host to be prepared.

On Linux, this command adds a new /dev/sg device file for each potential LUN for which a device file is not currently available.

On Solaris, this command adds entries to the file `/kernel/drv/sd.conf`, if necessary, for each potential new LUN that doesn't have an entry. It also generates an entry for each SCSI target to which the storage system is mapped. On Solaris 8, you must reboot the host after adding sd.conf entries. This command displays a warning whenever a reboot is necessary.

Setting up audit, recovery, and trace logging

Supported logs

SnapDrive for UNIX supports the following log files:

- Audit log. SnapDrive for UNIX logs all `snapdrive` commands and their return codes to an audit log. SnapDrive for UNIX makes an entry when you initiate a command and another when the command is complete. That entry includes both the status of the command and the completion time.
- Recovery log. Some SnapDrive for UNIX operations have the potential to leave the system in an inconsistent or less usable state if interrupted. This could happen if a user kills the program, or if the host crashes in the middle of an operation. The recovery log contains the steps of a `snap restore` operation. It documents the steps that were taken and the progress made. This way, technical support can assist you with the manual recovery process.
- Trace log. SnapDrive for UNIX reports information useful for diagnosing problems. If you have a problem, technical support might request this log file.

Enabling and disabling log files

To enable or disable each of these log files in the snapdrive.conf file, complete the following steps.

Step	Action
1	Log in as root.
2	Open the snapdrive.conf file in a text editor.

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="3">3</td><td>If you want to ...</td><td>Then ...</td></tr>
<tr><td>Enable a log file</td><td>Specify a file name as the value in the name/value pair of the log file you want to enable. SnapDrive for UNIX only writes log files if it has the name of a file to write to. The default names for the log files are as follows:
Audit log: sd-audit.log
Recovery log: sd-recovery.log
Trace log: sd-trace.log
Note
The path to these files may vary depending on your host operating system. For example, on AIX or HP-UX host, the default path for a log file is /var/snapdrive/...; on a Linux or Solaris host the default path is /var/log/... .</td></tr>
<tr><td>Disable a log file</td><td>Do not enter a value for the log file name parameter. If you do not supply a value, there is no file name to which SnapDrive for UNIX can write the log information.
Example: This example disables the audit log file.
<code>audit-log-file=""</code>
Note
NetApp strongly recommends that you leave all log files enabled.</td></tr>
<tr><td>4</td><td colspan="2">Save the file after you make all your changes.
SnapDrive for UNIX automatically checks this file each time it starts, so your changes take effect the next time it starts.</td></tr>
</table>

Settings affecting log file rotation

The values you specify in the snapdrive.conf file enable automatic log file rotations. You can change these values, if necessary, by editing the snapdrive.conf options. The following options affect log file rotation:

- audit-log-max-size
- audit-log-save
- trace-max-size
- trace-log-max-save
- recovery-log-save

Note

For information about the default values for these options, see "Determining options and their default values" on page 65.

With automatic log rotation, SnapDrive for UNIX keeps old log files until it reaches the limit specified in the <audit/trace/recovery>-log-save option. Then it deletes the oldest log file.

SnapDrive for UNIX tracks which file is oldest by assigning the file the number 0 when it creates the file. Each time it creates a new file, it increments by 1 the number assigned to each of the existing log files. When a log file's number reaches the save value, SnapDrive for UNIX deletes that file

Example: This example uses the `ls` command to produce information about the log files on the system.

Based on those settings, you would see the following information on log files:

```
# ls -l /var/log/sd*
-rw-r--r--   1 root     other      12247 Mar 13 13:09 /var/log/sd-audit.log
-rw-r--r--   1 root     other      20489 Mar 12 16:57 /var/log/sd-audit.log.0
-rw-r--r--   1 root     other      20536 Mar 12 03:13 /var/log/sd-audit.log.1
-rw-r--r--   1 root     other       3250 Mar 12 18:38 /var/log/sd-recovery.log.1
-rw-r--r--   1 root     other       6250 Mar 12 18:36 /var/log/sd-recovery.log.2
-rw-r--r--   1 root     other       6238 Mar 12 18:33 /var/log/sd-recovery.log.3
-rw-r--r--   1 root     other     191704 Mar 13 13:09 /var/log/sd-trace.log
-rw-r--r--   1 root     other     227929 Mar 12 16:57 /var/log/sd-trace.log.0
-rw-r--r--   1 root     other     213970 Mar 12 15:14 /var/log/sd-trace.log.1
-rw-r--r--   1 root     other     261697 Mar 12 14:16 /var/log/sd-trace.log.2
-rw-r--r--   1 root     other     232904 Mar 12 14:15 /var/log/sd-trace.log.3
-rw-r--r--   1 root     other     206905 Mar 12 14:14 /var/log/sd-trace.log.4
```

As shown in this example, the maximum size for each log indicates roughly what size it will be when SnapDrive for UNIX rotates it. SnapDrive for UNIX never rotates a log in the middle of an operation, so the sizes may vary from the default sizes.

Contents of an audit log

The audit log shows information about commands you issued with SnapDrive for UNIX. It maintains a history of the following information:

- The commands issued.
- The return value from those commands.
- The user ID of the user who invoked the command.
- A timestamp indicating when the command started (with no return code) and another timestamp indicating when the command finished (with a return code). The audit log record shows only information about `snapdrive` usage (issued commands).

An audit log file contains the following information.

Field	Description
uid	user ID
gid	group ID
msgText	message text
returnCode	return code from a command

Changing the defaults for audit logs

The snapdrive.conf file enables you to set the following parameters for audit logging.

Note

See "Determining options and their default values" on page 65 for more information about the default values for these options.

- The name of the file containing the audit log.
- The maximum size of the audit log file. The default size is 20K. After the file size reaches the value specified here, SnapDrive for UNIX renames the current audit log file by adding an arbitrary number to the name. Then it starts a new audit file using the name specified by the audit-log-file value.

- The maximum number of old audit files that Snapdrive saves. The default is 2.

Example of an audit log file

The following example shows an audit log file:

```
2501: Begin uid=0 gid=1 15:35:02 03/12/04 snapdrv snap create -dg
rdg -snapname snap_rdg1
2501: Status=0 15:35:07 03/12/04
2562: Begin uid=0 gid=1 15:35:16 03/12/04 snapdrv snap create -dg
rdg -snapname snap_rdg1
2562: FAILED Status=4 15:35:19 03/12/04
```

The first pair of lines in this example show an operation that succeeded, as indicated by the "Status=0" line.

The second pair of lines indicates an operation that failed. The return code of 4 means “already exists.” If you look at the two command lines, you can see that the first created a snapshot called snap_rdg1. The second line attempted to do the same, but the name already existed, so the operation failed.

Contents of a recovery log

If SnapDrive is halted using the Ctrl-C key sequence to kill the program, or if the host or storage system crashes in the middle of an operation, the system might not be able to recover automatically.

So, during any operation that, if interrupted, could leave the system in an inconsistent state, SnapDrive for UNIX writes information to a recovery log file. If a problem occurs, you can send this file to technical support so they can assist you in recovering the system’s state.

The recovery log utility records the commands that were issued in the process of the operation. Each is marked with an operation_index (a number that uniquely identifies the operation being executed), followed by the date/time stamp and the message text).

Changing the defaults for the recovery logs

The snapdrive.conf file enables you to set the following parameters for recovery logging.

Note

See “Determining options and their default values” on page 65 for more information about the default values for these options.

- The name of the file containing the recovery log, such as recoverylog.
- The maximum number of old recovery files that Snapdrive for UNIX saves. The default is 20. SnapDrive for UNIX keeps this number of recovery logs in case the problem with the process isn’t immediately discovered. SnapDrive for UNIX starts a new recovery log file each time it completes an operation. It renames the previous one by adding an arbitrary number to the name, such as recoverylog.0, recoverylog.1 and so on.

  Note

  The size of the recovery log file depends on the operation being performed. Each recovery log contains information about a single operation. When that operation is complete, SnapDrive for UNIX starts a new recovery log, regardless of how large the previous file was. As a result, there is no maximum size for a recovery log file.

Example of a recovery log file

The following is an example of entries in a recovery log where SnapDrive for UNIX has restored two snapshots before the operations halted. At this point, you would send this recovery log file to technical support for assistance in restoring the remaining snapshots.

```
6719: BEGIN 15:52:21 03/09/04 snapdrive snap restore -dg jssdg -
snapname natasha:/vol/vol0:abort_snap_restore
6719: BEGIN 15:52:27 03/09/04 create rollback snapshot:
natasha:/vol/vol0:abort_snap_restore.RESTORE_ROLLBACK_03092004_155
225
6719: END   15:52:29 03/09/04 create rollback snapshot:
natasha:/vol/vol0:abort_snap_restore.RESTORE_ROLLBACK_03092004_155
225 successful
6719: BEGIN 15:52:29 03/09/04 deactivate disk group:  jssdg
6719: BEGIN 15:52:29 03/09/04 stop host volume:
/dev/vx/dsk/jssdg/jvol_1
6719: END   15:52:30 03/09/04 stop host volume:
/dev/vx/dsk/jssdg/jvol_1 successful
6719: BEGIN 15:52:30 03/09/04 unmount file system:  /mnt/demo_fs
6719: END   15:52:30 03/09/04 unmount file system:  /mnt/demo_fs
successful
6719: BEGIN 15:52:30 03/09/04 stop host volume:
/dev/vx/dsk/jssdg/jvol_2
6719: END   15:52:30 03/09/04 stop host volume:
/dev/vx/dsk/jssdg/jvol_2 successful
6719: BEGIN 15:52:30 03/09/04 deport disk group:  jssdg
6719: END   15:52:30 03/09/04 deport disk group:  jssdg successful
6719: END   15:52:30 03/09/04 deactivate disk group:  jssdg
successful
6719: BEGIN 15:52:31 03/09/04 SFSR of LUN: /vol/vol0/lun1 from
snapshot: abort_snap_restore
6719: END   15:52:31 03/09/04 SFSR of LUN: /vol/vol0/lun1 from
snapshot: abort_snap_restore successful
6719: BEGIN 15:52:47 03/09/04 SFSR of LUN: /vol/vol0/lun2 from
snapshot: abort_snap_restore
6719: END   15:52:47 03/09/04 SFSR of LUN: /vol/vol0/lun2 from
snapshot: abort_snap_restore successful
```

About the trace log file

This file is for NetApp's use in cases where there is a problem that needs debugging. Enabling it does not affect system performance. By default, this file is enabled. You can disable it by setting the snapdrive.conf option trace-enabled to off.

Changing the defaults for the trace logs

The snapdrive.conf file enables you to set the following parameters for trace logging.

Note

See "Determining options and their default values" on page 65 for more information about the default values for these options.

- The name of the file containing the trace log.
- The maximum size of the trace log file. The default size is 0 bytes. This value ensures that each trace file will contain only one SnapDrive for UNIX command.

  If you reset the default size to a value other than 0, when the file reaches the size you specified, SnapDrive for UNIX renames the current trace log file by adding an arbitrary number to the name. Then it starts a new trace file using the name specified by the trace-log-file value.
- The maximum number of old trace files that SnapDrive for UNIX saves. The default is 20.
- The types of messages that SnapDrive for UNIX writes to the trace log file. By default, the trace log file contains fatal errors, admin errors, command errors, warnings and information messages.

Setting up autosupport

Understanding autosupport

NetApp provides autosupport with its storage systems as a way to provide better service to you, should you have a problem with your system. With autosupport, you can configure your storage system to send an e-mail message to technical support when an error occurs. Then, if you call in with an issue, technical support has information about your storage systems and configuration and can more quickly help you to solve the problem.

No secure information is ever sent using autosupport.

How SnapDrive for UNIX uses autosupport

For this release, SnapDrive for UNIX only sends an autosupport message from the storage system to NetApp the first time you execute it after a system reboot. It sends one message for each host reboot from the host that rebooted. At this time, it does not send a message when an error condition occurs.

Note

To use this feature, you must have a user login configured for the storage system and you must enable autosupport in the snapdrive.conf file. See "Enabling autosupport" on page 98

The autosupport feature in SnapDrive for UNIX logs into the storage system you configured for autosupport in the snapdrive.conf file. It uses that storage system to send an autosupport message to NetApp. This message specifies the following information:

- SnapDrive for UNIX version
- Message status: 3 for a warning or information message
- Host name
- Host operating system
- Host operating system release number
- Host operating system version

If the storage system specified for autosupport in the snapdrive.conf file cannot send an autosupport message to NetApp, SnapDrive for UNIX does not log an error to syslog. This information is only written to the internal SnapDrive for UNIX log files.

Examples of an autosupport message

The substance of the autosupport message is essentially the same regardless of your operating system. The details of the message, such as information on your operating system, vary according to your system setup.

HP-UX example: The following example is a message sent from a host named DBserver that is running release 5B.11.22 of HP-UX. This is an informational message, as indicated by the number 3 in parentheses (3).

```
snapdrive: 2.1 (3)
general: host_name = DBserver, host_os=HP-UX,
host_os_release=B.11.22, host_os_version=U
```

Linux example: The following example is a message sent from a host named DBserver that is running Linux. This is an informational message, as indicated by the number 3 in parentheses (3).

```
snapdrive: 2.1 (3)
general: host_name = DBserver, host_os=Linux,
host_os_release=2.4.21-9.ELsmp, host_os_version=#1 SMP Thu Jan 8
17:08:56 EST 2004.
```

Solaris example: The following example is a message sent from a host named DBserver that is running release 5.9 of Solaris. This is an informational message, as indicated by the number 3 in parentheses (3).

```
snapdrive: 2.1 (3)
general: host_name = DBserver, host_os=SunOS, host_os_release=5.9,
host_os_version=Generic
```

Enabling autosupport

To enable autosupport, complete the following steps.

Step	Action
1	Log in as root.
2	Set user logins for the storage systems you want SnapDrive for UNIX to access. See “Specifying the current login information for storage systems” on page 111.

Step	Action

3	Modify the snapdrive.conf file to enable autosupport.
	1. Set the autosupport-enabled parameter to 1. It should read as follows:
	`autosupport-enabled=1`
	2. Specify the name or IP address of a storage system that you want autosupport to contact and use for sending the messages. For example, if you want autosupport to use the storage system named toaster, the name/value pair would read as follows:
	`autosupport-filer=toaster`
	3. Save the snapdrive.conf file.
	SnapDrive for UNIX automatically checks this file each time it starts, so your changes take effect the next time it starts.

General steps for executing commands

What you can do with the commands

SnapDrive for UNIX supports snapshot management (`snap` commands) and storage provisioning (`storage` and `host` commands). In addition, you can use the `config` command to monitor or control some of SnapDrive for UNIX's configuration details.

Running SnapDrive for UNIX from the command line

You must be logged in as root to run SnapDrive for UNIX. You execute SnapDrive for UNIX from the CLI, either directly from a shell prompt or from scripts. When you execute SnapDrive for UNIX from the command line, keep the following in mind:

- Enter each command on a separate line.
- If you enter a command without arguments, SnapDrive for UNIX gives you a usage line with examples of commands.
- Each SnapDrive for UNIX command must use the following format:

  `snapdrive` *`type_name operation_name`* `[<`*`keyword/options`*`> <`*`arguments`*`>]`

  type_name specifies the type of task you want to perform. You can perform the following types of tasks:

  - Snapshot management. Use the type name `snap` with the `snapdrive` command to indicate you are performing a snapshot management operation.

    `snapdrive snap` *`operation <arguments>`*

  - Display storage provisioning and management information. Use the type name `storage` with the `snapdrive` command to indicate you are performing an operation dealing with storage on the storage system.

    `snapdrive storage` *`operation <arguments>`*

  - Connect or disconnect the host. Use the type name host with the snapdrive command to indicate you are performing an operation dealing only with the host-side entities. The host commands allow you to connect storage on the storage system to the host or disconnect that storage from the host. They do not affect the storage on the storage system. They affect only the visibility of the storage on the host.

    `snapdrive host` *`operation <arguments>`*

  - Specify or view behavioral options. Use the type name `config` with the `snapdrive` command to indicate you want to create, modify, or delete a

user login for a storage system. You also use the config type name to get SnapDrive for UNIX to view the snapdrive.conf file, view access permissions, or prepare the host for adding LUNs.

```
snapdrive config operation <arguments>
```

- Determine the version number of SnapDrive for UNIX.

```
snapdrive version <arguments>
```

operation_name specifies the operation you want to perform. Each command *type* has certain operations you can perform. The snapshot management operations are `create`, `rename`, `restore`, `show`, `connect`, `disconnect`, and `delete`. The storage operations are `create`, `connect`, `disconnect`, `resize`, `show`, and `delete`. The configuration operations are `access`, `show`, `list`, `set`, `delete`, and `prepare` and `check luns`. And the host operations are `connect` and `disconnect`.

<*keyword/options*> are the keywords that you can use to specify information corresponding to the host and storage system objects with which you are working, as well as options you can specify to get information such as verbose output. (See "SnapDrive for UNIX keywords and arguments" on page 292 for more information.)

<*arguments*> are the values you supply with the keywords that specify the target. (See "SnapDrive for UNIX keywords and arguments" on page 292 for a list of the acceptable arguments and the value you can supply for them.)

Frequently used command-line options

Some frequently used command-line options are the following:

- `-force` (`-f`). This option forces SnapDrive for UNIX to attempt an operation that it ordinarily would not undertake. SnapDrive for UNIX prompts you to confirm that you want it to perform this operation before it attempts the operation.
- `-noprompt`. This option keeps SnapDrive for UNIX from prompting you to confirm the operation. You should include this option if you're running SnapDrive for UNIX in a scripted environment where there is no person to confirm the operation.
- `-verbose` (`-v`). This option displays detailed, or verbose, output.

These options appear in the command examples in the following chapters.

Setting Up Security Features 4

About this chapter

This chapter contains information about the security features available in SnapDrive for UNIX and how to access them.

Topics in this chapter

This chapter discusses the following topics:

- "Setting up access control" on page 105
- "View the current access control settings" on page 109
- "Specifying the current login information for storage systems" on page 111
- "Enabling SSL encryption" on page 115

Overview of SnapDrive for UNIX's security features

Security features provided by SnapDrive for UNIX

SnapDrive for UNIX provides certain features to allow you to work with it in a more secure fashion. These features include letting you:

- Set up access control permissions
- Specify login information for the storage systems
- Specify that SnapDrive for UNIX use HTTPS (secure socket layer)

The access control feature lets you specify which operations a host running SnapDrive for UNIX can perform on a storage system. You set these permissions individually for each host. For more information, see "Setting up access control" on page 105.

In addition, to allow SnapDrive for UNIX to access a storage system, you must supply the login name and password for that storage system. For more information, see "Specifying the current login information for storage systems" on page 111.

The HTTPS feature lets you specify that you use it for all interactions with the storage system via the ManageONTAP interface, including sending the passwords.

Together these features give you more control over which users can perform operations on a storage system and from which host.

Setting up access control

Access control in SnapDrive for UNIX

SnapDrive for UNIX lets you control the level of access that each host has to each storage system to which the host is connected. This access level indicates which operations the host is allowed to perform when it targets a given storage system.

With the exception of the `show` and `list` operations, the access control permissions can affect all `snap` and `storage` operations.

Available access control levels

You can set the following access levels:

- NONE – The host has no access to the storage system.
- SNAP CREATE – The host can create snapshots.
- SNAP USE – The host can delete and rename snapshots.
- SNAP ALL – The host can create, restore, delete, and rename snapshots.
- STORAGE CREATE DELETE – The host can create, resize and delete storage.
- STORAGE USE – The host can connect and disconnect storage.
- STORAGE ALL – The host can create, delete, connect, and disconnect storage.
- ALL ACCESS– The host has access to all these SnapDrive for UNIX operations.

Each level is distinct. If you specify permission for only certain operations, then SnapDrive for UNIX can execute only those operations. For example, if you specify STORAGE USE, the host can use SnapDrive for UNIX to connect and disconnect to storage, but it cannot perform any other operations governed by access control permissions.

How access control works

SnapDrive for UNIX determines the access control permissions by checking the storage system for a permissions file in the root volume of the storage system. This file is in the directory sdprbac (SnapDrive Permissions Roles-Based Access Control). The file name is sd*host-name*.prbac, where *host-name* is the name of the host to which the permissions apply. You can have a permissions file for each host attached to the storage system.

Note
To determine the name of host, run the `hostname` command.

If the file is empty, unreadable, or has an invalid format, SnapDrive for UNIX does not grant the host access permission to any of the operations.

If the file is missing, SnapDrive for UNIX checks the variable all-access-if-rbac-unspecified in the snapdrive.conf file. If this variable is set to on (the default), it allows all hosts complete access to all these operations on that storage system. If it is set to off, SnapDrive for UNIX denies the host permission to perform any operations governed by access control on that storage system.

Setting access control permissions

To set the access control permissions for a storage system, complete the following steps.

Step	Action
1	Log in as root on the host.
2	On the storage system, create the directory sdprbac in the root volume of the target storage system. **Note** One way to make the root volume accessible is to mount the volume using NFS.
3	In this directory, create the permissions file. Make sure the following is true: ◆ The file must be named sd*host-name*.prbac where *host-name* is the name of the host for which you are specifying access permissions. ◆ The file must be read-only. This ensures that SnapDrive for UNIX can read it, but that it cannot be modified. **Example:** To give a host named dev-sun1 access permission, you would create the following file on the storage system. `/vol/vol0/sdprbac/sddev-sun1.prbac`

Step	Action

4	Set the permissions in the file for that host. You must use the following format for the file.
	◆ You can specify only one level of permissions. To give the host full access to all these operations, enter the string ALL ACCESS.
	◆ The permission string must be the first thing in the file. The file format is invalid if the permission string is not on the first line.
	◆ Permission strings are case insensitive.
	◆ No white space can precede the permission string.
	◆ No comments are allowed.
	The valid permission strings are:
	`NONE`
	`SNAP CREATE`
	`SNAP USE`
	`SNAP ALL`
	`STORAGE CREATE DELETE`
	`STORAGE USE`
	`STORAGE ALL`
	`ALL ACCESS`
	These strings allow the following access:
	◆ NONE – The host has no access to the storage system.
	◆ CREATE SNAP – The host can create snapshots.
	◆ SNAP USE – The host can delete and rename snapshots.
	◆ SNAP ALL– The host can create, restore, delete, and rename snapshots.
	◆ STORAGE CREATE DELETE – The host can create, resize, and delete storage.
	◆ STORAGE USE – The host can connect and disconnect storage.
	◆ STORAGE ALL – The host can create, delete, connect, and disconnect storage.
	◆ ALL ACCESS– The host has access to all the SnapDrive for UNIX operations.

Step	Action
	Each of these permission strings is discrete. If you specify SNAP USE, the host can delete or rename snapshots, but it cannot create snapshots or restore snapshots or perform any storage provisioning operations. Regardless of the permissions you set, the host can perform show and list operations.
5	Verify the access permissions by entering the following command: `snapdrive config access show` *`filer-name`*

View the current access control settings

About viewing the access control permissions

You use the `snapdrive config access` command to display information about the permissions available for a host on a specific storage system.

Viewing the access control permissions

To view the access control permissions, complete the following step.

Step	Action

1	Execute the `snapdrive config access show` command. This command has the following format:
	`snapdrive config access {show | list} filer_name`
	You can use the same arguments regardless of whether you enter the `show` or `list` version of the command.
	Note
	For details about using the options and arguments in this command line, see the section "SnapDrive for UNIX keywords and arguments" on page 292.
	Result: This operation displays information about the access permissions available for that host.
	Example 1: This command line checks the storage system toaster to determine which permissions the host has. Based on the permissions shown, the permissions for the host on this storage system are SNAP ALL.
	`# snapdrive config access show toaster`
	`This host has the following access permission to filer, toaster:`
	`SNAP ALL`
	`Commands allowed:`
	`snap create`
	`snap restore`
	`snap delete`
	`snap rename`
	`#`

Step	Action

	Example 2: In this example, the permissions file is not on the storage system, so SnapDrive for UNIX checks the variable all-access-if-rbac-unspecified in the snapdrive.conf file to determine which permissions the host has. This variable is set to on, which is equivalent to creating a permissions file with the access level set to ALL ACCESS.
	
	`# snapdrive config access list toaster`
	`This host has the following access permission to filer, toaster:`
	`ALL ACCESS`
	`Commands allowed:`
	`snap create`
	`snap restore`
	`snap delete`
	`snap rename`
	`storage create`
	`storage resize`
	`snap connect`
	`storage connect`
	`storage delete`
	`snap disconnect`
	`storage disconnect`
	`#`
	
	Example 3: This example shows the kind of message you would receive if there is not a permissions file on the storage system toaster and the variable all-access-if-rbac-unspecified in the snapdrive.conf file is set to off.
	
	`# snapdrive config access list toaster`
	`Unable to read the access permission file on filer, toaster. Verify that the file is present.`
	`Granting no permissions to filer, toaster.`
	`#`

Specifying the current login information for storage systems

About using current logins for storage systems

A user name/password enables SnapDrive for UNIX to access each storage storage system. It also provides a level of security because, in addition to being logged in as root, the person running SnapDrive for UNIX must supply the correct user name/password when prompted for it. If a login is compromised, you can delete it and set a new user login.

You created the user login for each storage system when you set it up. For SnapDrive for UNIX to work with the storage system, you must supply it with this login information. Depending on what you specified when you set up the storage systems, each storage system could use either the same login or a unique login.

SnapDrive for UNIX stores these logins and passwords in encrypted form on each host. You can specify that it encrypt this information when you send it across the wire by setting the snapdrive.conf variable `use-https-to-filer=on`. For more information, see "Enabling SSL encryption" on page 115.

Specifying login information

To specify the user login information for a storage system, complete the following steps.

Note

Depending on what you specified when you set up the storage system, each storage system could use either the same user name/password or a unique user name/password. If all the storage systems use the same user name/password information, you only need to perform the following steps once. If not, repeat these steps for each storage system.

Step	Action
1	Log in as root.

Step	Action
2	Enter the following command: `snapdrive config set` *`user_name filername`* `[`*`filername`* `...]` *user_name* is the user name that was specified for that storage system when you first set it up. If you don't include a value for it, SnapDrive for UNIX prompts you for one. *filername* is the name of the storage system. You can enter multiple storage system names on one command line if they all have the same user login/password. You must enter the name of at least one storage system.

3	At the prompt, enter the password, if there is one.
	Note
	If no password was set, press Enter (i.e., the null value) when prompted for a password.
	Example 1: This example sets up a user called root for a storage system called toaster.
	`# snapdrive config set root toaster`
	`Password for root:`
	`Retype Password:`
	`Access to filer toaster has been configured for user root`
	Example 2: This example sets up one user called root for three storage systems.
	`# snapdrive config set root toaster oven broiler`
	`Password for root:`
	`Retype Password:`
	`Access to filers toaster, oven, and broiler has been configured for user root`

4	If you have another storage system with a different user name/password, repeat these steps.

Verifying storage system user names associated with SnapDrive for UNIX

You can verify which user name SnapDrive for UNIX has associated with a storage system by executing the `snapdrive config list` command.

Note

This command does not query the storage system to determine whether addition user names have been configured for it. Nor does it display the password associated with a storage system.

To execute `config list`, complete the following step.

Step	Action
1	Log in as root.

2	Enter the following command:
	`snapdrive config list`
	This command displays the user name/storage system pairs for all systems that have users specified within SnapDrive for UNIX. It does not display the passwords for the storage systems.
	Example: This example displays the users associated with the storage systems rapunzel and mediumfiler.
	`# snapdrive config list`
	`user name filer name`
	`----------`
	`rumplestiltskins rapunzel`
	`longuser mediumfiler`

Deleting a user login for a storage system

To delete a user login for one or more storage systems, complete the following steps.

Step	Action
1	Log in as root.

Step	Action
2	Enter the following command: `snapdrive config delete filername [filername ...]` *filername* is the name of the storage system for which you want to delete the user login information. **Result:** SnapDrive for UNIX removes the user name/password login information for the storage systems you specify. **Note** To enable SnapDrive for UNIX to access the storage system, you must specify a new user login.

Enabling SSL encryption

Enabling SnapDrive for UNIX to use HTTPS

You can set up SnapDrive for UNIX to use HTTPS (secure socket layer encryption) when it sends information about storage system user names and passwords across the network.

SnapDrive for UNIX uses HTTP instead of HTTPS as its default when it sends storage system login information across the network. By changing the `use-https-to-filer` variable in the snapdrive.conf file, you can tell SnapDrive for UNIX to use HTTPS (SSL encryption) to communicate with the storage system.

Note

Using HTTPS might result in slower performance with versions of Data ONTAP prior to 7.0. It does not affect performance with Data ONTAP. 7.0.

Setting up HTTPS

To set up SnapDrive for UNIX to use HTTPS, complete the following steps:

Note

For more information on the configuration file, see "Setting configuration information" on page 64.

Step	Action
1	Log on as root.
2	Make a backup of the snapdrive.conf file.
3	Open the snapdrive.conf file in a text editor.

Step	Action
4	Change the value for the `use-https-to-filer` variable to on. `use-https-to-filer variable=on` **Tip:** A good practice any time you modify the snapdrive.conf file is to perform the following steps: **1.** Comment out the line you want to modify. **2.** Copy the commented-out line. **3.** Un-comment the copy by removing the pound sign. **4.** Modify the value. This way you always have a record of the default value in the file.
5	Save the file after you make your changes. SnapDrive for UNIX automatically checks this file each time it starts, so your changes take effect the next time it starts.

Creating and Using Snapshots 5

About this chapter

This chapter provides details about creating and using SnapDrive for UNIX snapshots.

Topics in this chapter

This chapter discusses the following topics:

- "Creating snapshots" on page 118
- "Displaying information about snapshots" on page 127
- "Renaming a snapshot" on page 134
- "Restoring a snapshot" on page 136
- "Connecting to a snapshot" on page 147
- "Disconnecting a snapshot" on page 160
- "Deleting snapshots" on page 165

Creating snapshots

About creating snapshots

You use the `snapdrive snap create` command to create snapshots, which are point-in-time, read-only images of data on storage system volumes. The `snap create` operation ensures that you have backed up your LUNs or NFS files and directory trees. You can use the snapshot you create to restore your data if you encounter corruption or other problems.

Crash-consistent snapshots

When you create a snapshot of a storage entity, like a file system or disk group, Snapdrive creates a snapshot that contains the image of all the storage system volumes that comprise the entity you specified using a *file_spec* argument. The file_spec argument specifies the storage entity, like the file system, LUN, or NFS directory tree, that SnapDrive uses to create the snapshot.

SnapDrive makes consistent the storage components that comprise the entity you requested in the snapshot. This means that LUNs or directories being used outside those specified by the `snap create` command's *file_spec* argument may not have consistent images in the snapshot. SnapDrive for UNIX enables you to restore only the entities specified by the *filespec* argument that are made consistent in the snapshot.

Snapshots of entities contained on a single storage system volume are always crash-consistent. SnapDrive for UNIX takes special steps to ensure that snapshots that span multiple storage systems or storage system volumes are also crash-consistent. When you create a snapshot that spans multiple storage system volumes, SnapDrive for UNIX ensures consistency by freezing I/O to the requested LUNs. If a freeze is not permitted, as for example on a Linux host, SnapDrive for UNIX makes a "best effort" to create a consistent snapshot by taking the snapshot without freezing the target storage, and then checking for read-write I/Os that occurred to the storage entities when the snapshot was taken. If SnapDrive can create a crash-consistent snapshot, the `snap create` command succeeds. If it can not, SnapDrive discards the snapshot and informs the user of the failure. In some instances, SnapDrive for UNIX can not create the snapshot. SnapDrive will never create a snapshot unless the data is crash-consistent.

The following table shows the host systems and snapshot entities where SnapDrive for UNIX can guarantee a crash-consistent snapshot. You can sometimes create crash-consistent snapshots in configurations that are not guaranteed, but this requires additional steps and may also require multiple attempts, especially if the target storage is under load. You must perform

whatever steps are necessary to quiesce the application before taking a snapshot on a configuration where snapshots are not guaranteed. Note that database hot backup facilities depend on the methods used by the DBMS, and do not always quiesce I/O to database files.

Host	Snapshot entities that span multiple volumes				
	LVM file systems	LVM host volume or disk group	File system on LUN (two or more)	LUN (two or more)	NFS file or directory tree (two or more)
Solaris	Guaranteed	Best Effort	Guaranteed	Best Effort	Best Effort
HP-UX	Guaranteed	Best Effort	Guaranteed	Best Effort	Best Effort
AIX	Guaranteed	Guaranteed	Best Effort	Best Effort	Best Effort
Linux	Best Effort	Best Effort	Best Effort	Best Effort	Best Effort

Application consistent snapshots

To ensure that a snapshot is "application-consistent," you may need to stop or do whatever steps are required to quiesce the application before taking the snapshot. Note that database hot backup facilities depend on the methods used by the DBMS, and do not always quiesce I/O to database files.

If the application has not completed its transactions and written data to the storage system, the resulting snapshot might not be application consistent.

Note

If your application can recover from a crash-consistent snapshot, you may not need to stop it. Consult the documentation for your application. For more information on taking application consistent snapshots, see "Architectural guidelines for SnapDrive for UNIX" on page 7.

You should take a new snapshot whenever you add or remove a host volume, LUN, or NFS directory tree, or resize host volumes or file systems. This ensures that you have a consistent copy of the newly configured disk group that you can use if you need to restore the disk group.

Snapshots that span storage systems, volumes

SnapDrive allows you to take snapshots that span multiple storage system volumes or multiple storage systems. These volumes can reside on the same storage system or different storage systems. Although the `snap create`

command creates a snapshot of all the volumes that comprise the entity you request, SnapDrive for UNIX will restore only the entities that you specify in the `snap create` command.

When you use the `snap create` command to take a snapshot that spans multiple volumes, you do not need to name the volumes on the command line. SnapDrive for UNIX gets this information from the *file_spec* argument that you specify.

- If the *file_spec* you enter requests a disk group, or a file system or host volume that resides on a disk group, SnapDrive automatically creates a snapshot that includes all the storage system volumes for the disk group, volume, or file system you specified.
- If the *file_spec* you enter requests a LUN, SnapDrive for UNIX takes a snapshot of the storage system volume that contains the LUN.
- If the *file_spec* you enter requests a file system that resides directly on a LUN, SnapDrive for UNIX takes a snapshot the storage system volume that contains the LUN and file system that you specified.
- If the *file_spec* you enter requests an NFS directory, SnapDrive creates a snapshot of the volume that contains the NFS directory tree.

In addition to using a *file_spec* argument that is built on entities from multiple storage systems and storage system volumes, you can also use a combination of *file_spec* arguments where each value is based on single storage system or storage system volume. For example, suppose you have a setup where the disk group dg1 spans the storage systems filer2 and filer3, dg2 is on filer2, and dg3 is on filer3. In this case, any of the following command lines would be correct:

```
snap create -dg dg1 -snapname snapdg1
snap create -dg dg2 dg3 -snapname snapdg23
snap create -dg dg1 dg2 dg3 -snapname snapdg123
```

Something to keep in mind when creating snapshots that span storage systems and volumes is that SnapDrive for UNIX creates the snapshot on each storage systems volume using a short name. It does not include the full pathname in the name, even if the volumes are on different storage system. This means that if you later rename the snapshot, you must go to each storage system and volume and rename it there as well.

Creating snapshots of unrelated entities

Unless you specify otherwise, SnapDrive for UNIX assumes that all entities that you specify on a given `snap create` command line are related, in other words that the validity of updates to one entity may depend on updates to the other

entities specified. When storage entities have "dependent writes" in this way, SnapDrive for UNIX takes steps to create a snapshot that is crash-consistent for all storage entities as a group.

The following example shows how SnapDrive for UNIX creates a snapshot of storage entities that may have dependent writes. In the example below, the `snap create command` specifies a file system on a LUN and also a disk group. The disk group consists of LUNs residing on a single storage system volume. The file system on a LUN resides on a different storage system and storage system volume. As a group, the file system and the disk group span multiple storage system volumes. Individually they do not.

The following command specifies a snapshot that contains both the file system `/mnt/fs1` and the disk group `dg1`.

```
snapdrive snap create -fs /mnt/fs1 -dg dg1 -snapname fs1_dg1
```

Because these storage entities may have dependent writes, SnapDrive for UNIX attempts to create a crash-consistent snapshot, and treats the file system /mnt/fs1 and the disk group dg1 as a group. This means SnapDrive is required to freeze I/O operations to the storage system volumes before creating the snapshot.

Creating crash-consistent snapshots for multiple storage entities across volumes takes extra time, and is not always possible if SnapDrive cannot freeze I/O operations. Because this is so, SnapDrive allows you to create snapshots of “unrelated” storage entities. Unrelated storage entities are entities that you can specify that have no dependent writes when the snapshot is taken. Because the entities have no dependent writes, SnapDrive does not take steps to make the entities consistent as a group. Instead, SnapDrive creates a snapshot in which each of the individual storage entities is crash-consistent.

The following command specifies a snapshot of the file system on a LUN and the disk group described previously. Because the `-unrelated` option is specified, SnapDrive creates a snapshot in which the file system /mnt/fs1 and the disk group dg1 are crash-consistent as individual storage entities, but are not treated as a group. The following command does not require SnapDrive to freeze I/O operations on the storage system volumes.

```
snapdrive snap create -fs /mnt/fs1 -dg dg1 -unrelated -snapname
fs1_dg1
```

Refer to “Crash-consistent snapshots” on page 118 for additional information about how SnapDrive ensures crash consistency.

Guidelines for snapshot creation

Follow these guidelines when you enter commands that create snapshots:

- You can keep a maximum of 255 snapshots per storage system volume. This limit is set by the storage system. The total number can vary depending on whether other tools use these snapshots.

  When the number of snapshots has reached the maximum limit, the `snapshot create` operation fails. You must delete some of the old snapshots before you can use SnapDrive for UNIX to take any more.
- SnapDrive for UNIX does not support snapshots that it does not create. For example, it does not support snapshots that are created from the storage system console, because such a practice can lead to inconsistencies within the file system.
- You cannot use SnapDrive for UNIX to create snapshots of the following:
  - Root disk groups. The `snap create` operation fails when you try to take a snapshot of a root disk group for a logical volume manager.
  - Boot device or swap device. SnapDrive for UNIX does not take a snapshot of a system boot device or a system swap device.
  - Shared disk group. SnapDrive for UNIX does not create snapshots of disk groups that are shared with a host cluster partner.
- When a snapshot spans multiple storage systems or storage system volumes, SnapDrive for UNIX requires a freeze operation to guarantee crash-consistency. Refer to "Crash-consistent snapshots" on page 118, for information about creating snapshots on configurations where a freeze is not provided.

Creating a snapshot

To create a snapshot using SnapDrive for UNIX, complete the following steps.

Note

SnapDrive for UNIX can only restore snapshots it takes.

Step	Action

1	Execute the `snapdrive snap create` command. This command has the following format:
	
	`snapdrive snap create`
	`{-lun | -dg | -fs | -hostvol }` *`file_spec`* `[`*`file_spec`* `...]`
	`[ {-lun |-dg | -fs | -hostvol }` *`file_spec`* `[`*`file_spec`*`...]]`
	`-snapname` *`snap_name`* `[ -force [-noprompt]][-unrelated]`
	
	Note
	The `-dg` and `-vg` options are synonyms that reflect the fact that some operating systems refer to disk groups and others refer to volume groups. In addition, `-lvol` and `-hostvol` are also synonyms. This guide uses `-dg` to refer to both disk groups and volume groups and `-hostvol` to refer to both logical volumes and host volumes. For details about using all the options and arguments available with this command, see "SnapDrive for UNIX keywords and arguments" on page 292.
	
	Examples of possible command lines are included at the end of this table.
	
	Keep the following in mind as you create snapshots:
	◆ *file_spec* is the name of the storage entity you specify. For example, if you use the `-dg` option, *file_spec* is the name of that disk group. If you specify `-fs`, *file_spec* is the name of the NFS or file system mount point. You can specify NFS entities, LUNs, file systems created directly on LUNs and LVM entities on the same command line.
	The value you supply for an *file_spec* argument can span multiple storage systems and storage system volumes.

Step	Action
	◆ If you want SnapDrive for UNIX to overwrite an existing snapshot, you must include the `-force` option on the command line. Otherwise, the `create` operation fails. (If the old snapshot is in use, this operation fails even with the `-force` option.) The `-force` option first asks whether you want to overwrite the existing snapshot. To avoid getting this prompt, include the `-noprompt` option on the command. When you include this option with the `-force` option, SnapDrive for UNIX performs the operation without checking with you. ◆ If you want to use the same *file_spec* argument multiple times in the same command line, enter the option once and follow it by multiple arguments. For example, if you want to specify multiple disk groups, enter `-dg` once and list all disk groups after it. So you have the argument `-dg dg1 dg2`. ◆ If you use the `-hostvo`l option, make sure that it specifies both the requested host volume and the disk group containing it, such as `-hostvol dg3/acct_volume`. ◆ The `-unrelated` option creates a snapshot of *file_spec* entities that have no dependent writes when the snapshot is taken. Because the entities have no dependent writes, SnapDrive creates a crash-consistent snapshot of the individual storage entities, but does not take steps to make the entities consistent as a group. ◆ If you use the `-lun` option, you must specify the long snapshot name. ◆ All the LUNs in the disk group you are taking a snapshot of must be online and available. ◆ Whenever you add or remove a host volume, LUN, or NFS directory tree, or resize host volumes or file systems, you should take a new snapshot. This ensures that you have a consistent copy of the storage entity to use for the `restore` operation when you need it. **Result:** The *file_spec* arguments represent a set storage entities on one or more storage systems. The `snap create` operation takes a snapshot of the storage system volume containing those entities and gives it the name specified in the *snap_name* argument.

Step	Action

	Examples of command lines: These examples use a host called DBserver. The disk group dg1 has host volumes myvol1 and myvol2. The host volume dg1/myvol2 has a file system mounted on /fs2. The disk group dg1 has three LUNs in it: toaster:/vol/vol1/lun0, toaster:/vol/vol1/lun1, and toaster: /vol/vol1/lun2. The following command lines are all valid, and back up the same data. These command lines fall into three general categories. The first command line uses a file system; the next to refer to disk/volume groups, and the last two refer to host volumes/logical volumes. All the command lines create a snapshot called toaster:/vol/vol1:snap1.
	`snapdrive snap create -fs /fs2 -snapname snap1`
	`snapdrive snap create -dg dg1 -snapname snap1`
	`snapdrive snap create -vg dg1 -snapname snap1`
	`snapdrive snap create -hostvol dg1/myvol1 -snapname snap1`
	`snapdrive snap create -lvol dg1/myvol2 -fs /fs2 -snapname snap1`
	Example 2: This example creates a multivolume snapshot for a Linux host. The snapshot contains the disk group vgmultivol, which include the host volumes lvol1 and lvol2.
	`# snapdrive snap create -vg vgmultivol -snapname snapmultivol`
	`Successfully created snapshot snapmultivol on 2 filer volumes:`
	`toaster:/vol/vol1`
	`toaster:/vol/vol2`
	`snapshot snapmultivol contains:`
	`disk group vgmultivol containing host volumes`
	`lvol1`
	`lvol2`
	`#`

Step	Action

	Example 3: These examples create snapshots of file systems that are created directly on LUNs.
	
	The following creates a snapshot of the file system mounted at /mnt/fs1. The snapshot contains the LUN on which the file system is created.
	
	`# snapdrive snap create -fs /mnt/fs1 -snapname snapfs1`
	
	The next example creates a snapshot of multiple file systems. When the file system spans storage systems or storage system volumes, SnapDrive automatically locates the storage system volumes that contain the LUNs with the file systems.
	
	`snapdrive snap create -fs /mnt/fs1 /mnt/fs2 -snapname snapfs12`
	
	Example 4: This example creates a snapshot of LUNs that are mapped directly to the host. The `-lun` specification must include the storage system, volume and LUN name.
	
	`snapdrive snap create -lun ham:/vol/vol1/luna lunb -snapname twoluns`
	
	Example 5: This example creates a snapshot of LUNs that reside on multiple storage systems.
	
	`snapdrive snap create -lun filer1:/vol/vol1/luna filer2:/vol/vol1/lunb -snapname lunsab`
	
	Example 6: This example creates a snapshot of storage system entities that do not have dependent writes during snapshot creation. SnapDrive for UNIX creates a snapshot in which the file system /mnt/fs1 and the disk group dg1 are crash-consistent as individual storage entities, but are not treated as a group.
	
	`snapdrive snap create -fs /mnt/fs1 -dg dg1 -unrelated -snapname fs1_dg1`

Displaying information about snapshots

Command to use to display snapshot information

You can use the `snapdrive snap show` (or `list`) command to display information about each snapshot taken by SnapDrive for UNIX. You can use this command to display information on the following:

- Storage systems
- Volumes on storage systems
- Storage entities such as NFS files and directory trees, volume groups, disk groups, file systems, logical volumes, and host volumes
- Snapshots

Displaying information about a snapshot

To display snapshot information, complete the following step.

Step	Action

1	Execute the `snapdrive snap show` command. This command has the following formats:
	`snapdrive snap { show | list } -filer` *`filername`* `[`*`filername`*`...] [-verbose]`
	`snapdrive snap { show | list } -filervol` *`filervol`* `[`*`filervol`*`...] [-verbose]`
	`snapdrive snap { show | list } {- lun | -dg | -fs | -hostvol }` *`file_spec`* `[`*`file_spec`* `...] [-verbose]`
	`snapdrive snap { show | list } [-snapname]` *`long_snap_name`* `[`*`snap_name`* `...] [-verbose]`
	Note
	You can use the same arguments regardless of whether you enter `snapdrive snap show` or `snapdrive snap list` as the command. These commands a synonyms. For similar reasons, the `-dg` and `-vg` options are synonyms that reflect the fact that some operating systems refer to disk groups and others refer to volume groups. In addition, `-lvol` and `-hostvol` are also synonyms. This guide uses `-dg` to refer to both disk groups and volume groups and `-hostvol` to refer to both logical volumes and host volumes. For details about using all the options and arguments available with this command, see "SnapDrive for UNIX keywords and arguments" on page 292.

Step	Action
	◆ You can use this command to display information about a storage system entity, such as storage system (-filer), storage system volume (-filervol), or snapshot (`-snapname`). You can also combine these options with options as specified by the *file_spec* argument. For example, you could find all snapshots that have disk group dg1 on the storage system comet even though dg1 doesn't exist on your host by running one command: `snap show -dg dg1 -filer comet` ◆ You can use the wildcard (*) character in snapshot names. The `snap show` operation lets you use the wildcard character to show all snapshot names that match a certain pattern or all the snapshots on a particular volume. The following rules apply to using wildcards in snapshot names: ❖ You can use a wildcard at the end of the name only. You cannot use the wildcard at the beginning or the middle of a snapshot name. ❖ You cannot use the wildcard in the storage system or storage system volume fields of a snapshot name. See the examples that follow for a sample command line that includes a wildcard. **Result:** This operation displays, at a minimum, the following information about the snapshot: ◆ The name of the storage system where the snapshot was taken ◆ The name of the host that took the snapshot ◆ The path to the LUNs on the storage system ◆ The date and time the snapshot was taken ◆ The name of the snapshot ◆ The names of the disk groups included in the snapshot You can also use this command to list of all snapshots on specific objects, including storage systems and their volumes, disk groups, host volume groups, file systems, host volumes, and logical volumes. When you include the verbose option, SnapDrive for UNIX displays more information. If you enter a `snap show` command and SnapDrive for UNIX doesn't locate any snapshots, it displays the message "no matching snapshots." If you specify arguments on the command line, and some portion of them do not exist, SnapDrive for UNIX returns a partial listing of those for which snapshots are found. It also lists the arguments that were invalid.

Step	Action

	Example 1: The following are examples of `snapdrive snap show` commands:
	`snapdrive snap show -snapname toaster:/vol/vol2:snapA snapX snapY`
	`snapdrive snap show -verbose toaster:/vol/vol2:snapA /vol/vol3:snapB snapC`
	`snapdrive snap show toaster:/vol/vol2:snapA`
	`snapdrive snap list -dg dg1 dg2`
	Example 2: This example uses a wildcard to display information about the available snapshots on a particular volume:
	`# snapdrive snap show toaster:/vol/vol0:*`
	`snap name host date snapped`
	`----------`
	`toaster:/vol/vol0:sss1 DBserver Mar 12 16:19 dg1`
	`toaster:/vol/vol0:testdg DBserver Mar 12 15:35 dg1`
	`toaster:/vol/vol0:t1 DBserver Mar 10 18:06 dg1`
	`toaster:/vol/vol0:hp_1 HPserver Mar 8 19:01 vg01`
	`toaster:/vol/vol0:r3 DBserver Mar 8 13:39 rdg1`
	`toaster:/vol/vol0:r1 DBserver Mar 8 13:20 rdg1`
	`toaster:/vol/vol0:snap2 DBserver Mar 8 11:51`
	`rdg1toaster:/vol/vol0:snap_p1 DBserver Mar 8 10:18 rdg1`
	Example 3: In this example, the verbose option is included on the command line.
	`# snapdrive snap show -v toaster:/vol/vol0:snap_p1`
	`snap name host date snapped`
	`----------`
	`toaster:/vol/vol0:snap_p1 DBserver Mar 8 10:18 rdg1`
	`host OS: SunOS 5.8 Generic_108528-23`
	`snapshot name: snap_p1`
	`Volume Manager: vxvm 3.5`
	`disk group: rdg`
	`host volume: nvol1`
	`file system: nvol1 mountpoint: /mnt/rdg_nvol1`
	`lun path`
	`----------`
	`toaster:/vol/vol0/rds0`
	`lun path`
	`----------`
	`myfiler:/vol0/lun-1`
	`myfiler:/vol0/lun-1`
	`myfiler:/vol0/lun-5`
	`myfiler:/vol0/lun-5`

Step	Action

	Example 4: This example on an AIX host includes messages about snapshots that were not created with SnapDrive for UNIX.
	
	`# snapdrive snap show -filer toaster`
	`snap name  host  date  snapped`
	`----------`
	`toaster:/vol/vol0:hourly.0  non-snapdrive snapshot`
	`toaster:/vol/vol1:hourly.0  non-snapdrive snapshot`
	`toaster:/vol/vol1:snap1  DBserver1  Oct 01 13:42  dg1 dg2`
	`toaster:/vol/vol1:snap2  DBserver2  Oct 10 13:40  DBdg/hvol1`
	`toaster:/vol/vol1:snap3  DBserver3  Oct 31 13:45  DBdg`
	
	Example 5: This example displays a snapshot of an LVM-based file system on an AIX host using the `snapdrive snap show` command and the verbose option.
	`# ./snapdrive snap show -v -fs /mnt/check_submit/csdg2/hv2_1`
	
	`snapname  host  date  snapped`
	`----------`
	`toaster:/vol/vol1:mysnapshot sales-aix  Aug 24 10:55  csdg2`
	
	`host OS: AIX 1 5`
	
	`snapshot name: mysnapshot`
	
	`Volume Manager: aixlvm 5.1`
	
	`disk group: csdg2`
	`host volume: csdg2_log`
	`host volume: csdg2_hv2_1`
	`file system: csdg2_hv2_1 file system type: jfs2 mountpoint:`
	`/mnt/check_submit/csdg2/hv2_1`
	
	`lun path dev paths`
	`----------`
	`spinel:/vol/vol1/check_submit_aix-4 /dev/hdisk4`

Step	Action

	Example 6: This example shows a snapshot of an NFS-mounted directory tree on a Linux host using the `snapdrive snap list` command with the verbose option.
	
	`# snapdrive snap list -fs /mnt/acctfs1 -v`
	
	`snap name host date snapped`
	`----------`
	`besser:/vol/vol1:acctfs-s1 childs Aug 8 18:58 /mnt/acctfs1`
	
	`host OS: Linux 2.4.21-9.ELsmp #1 SMP Thu Jan 8 17:08:56 EST 2004`
	
	`snapshot name: acctfs-s1 file system: type: nfs mountpoint: /mnt/acctfs1`
	`filer dir: besser:/vol/vol1`

Step	Action

	Example 7: This example executes the `snap show` command on a Linux host.
	
	`# snapdrive snap show -snapname surf:/vol/vol1:swzldg5snapped`
	
	`snap name host date snapped`
	`-------------------------------------------------------------------------------`
	`surf:/vol/vol1:bagel5snapped pons Aug 18 20:06 dg5`
	`#`
	`# ./linux/ix86/snapdrive snap show -v -snapname surf:/vol/vol1:bagel5snapped >`
	
	`snap name host date snapped`
	`-------------------------------------------------------------------------------`
	`surf:/vol/vol1:bagel5snapped pons Aug 18 20:06 dg5`
	
	`host OS: Linux 2.4.21-9.ELsmp #1 SMP Thu Jan 8 17:08:56 EST 2004`
	
	`snapshot name: bagel5snapped`
	
	`Volume Manager: linuxlvm 1.0.3`
	
	`disk group: dg5`
	`host volume: vol1`
	`host volume: vol2`
	`host volume: vol3`
	
	`lun path dev paths`
	`-----------------------------------------------------`
	`surf:/vol/vol1/glk19 /dev/sdu`
	`#`
	
	Examples with wildcards: The following examples use wildcards:
	
	`snapdrive snap show toaster:/vol/volX:*`
	`snapdrive snap show -v toaster:/vol/volX:DB1* filer1:/vol/volY:DB2*`
	`snapdrive snap show toaster:/vol/vol2:mysnap* /vol/vol2:yoursnap* hersnap*`
	`snapdrive snap show toaster:/vol/volX:*`
	
	Example of an invalid command: The following use of a wildcard is invalid because the wildcard is in the middle of the name instead of at the end:
	
	`snap show toaster:/vol/vol1:my*snap`

Other ways to get snapshot names

Another way to get a snapshot name is to log on to the storage system and use the `snap list` command there. This command displays the names of the snapshots.

Note

The `snapdrive snap show` command is equivalent to the storage system `snap list` command.

Renaming a snapshot

Command to use to rename a snapshot

You can use the `snapshot snap rename` command to change the name of an existing snapshot.

Renaming a snapshot that spans systems, volumes

You can also use this command to rename a snapshot that is across multiple storage systems or multiple storage system volumes.

If you rename one of these snapshots, you must also rename all the related snapshots using the same name. This is because SnapDrive for UNIX uses a short name when it creates the snapshot, even though it spans multiple storage systems or volumes. The rename command changes the name of the current snapshot but it does not change the name of the related snapshots in the other locations.

Changing a snapshot name

To change the name of a snapshot, complete the following step.

Step	Action
1	Execute the `snapdrive snap rename` command. This command has the following format: `snapdrive snap rename [-snapname] old_long_snap_name new_snap_name [-force [-noprompt]]` *old_long_snap_name* is the current name of the snapshot. It must use the long format of the name where you enter the storage system name, volume, and snapshot name. *new_snap_name* is the name you want to give the snapshot. You can use the short form of the name for the *new_snap_name* because it must exist on the same storage system volume as the original snapshot. An error occurs if you try to rename a snapshot to a different storage system. If you created a snapshot that spans multiple storage systems or storage system volumes, make sure you rename each related snapshot with the same name.

Step	Action

	If the new snapshot name exists, you must use the `-force` option on the command line to force SnapDrive to use this name. Otherwise, the operation fails. This option prompts you to confirm that you want to use the existing name. To prevent SnapDrive for UNIX from displaying the prompt, include the `-noprompt` option also.
	
	Note
	The section "SnapDrive for UNIX keywords and arguments" on page 292 provides details about using the options and arguments in this command line.
	
	Result: The `snap rename` operation changes the name of the source snapshot source to the name specified by the target argument.
	
	Example: The following are examples of the `snapdrive snap rename` command. The first command line includes the `-force` option because a snapshot named newsnapshot already exists. In the second example, both snapshot names use the long form of the name, but they both resolve to the same storage system volume.
	
	`snapdrive snap rename -force filer1:/vol/vol1:oldsnap newsnapshot`
	
	`snapdrive snap rename filer1:/vol/vol1:FridaySnap filer1:/vol/vol1:Snap040130`

Restoring a snapshot

Command to use to restore snapshots

The `snapdrive snap restore` command restores data from the snapshot you specify on the command line to the storage system. This operation replaces the contents of the *file_spec* arguments (for example disk groups, NFS files, NFS directory trees, file systems created directly on LUNs) that you specified on the `snap restore` command line with the contents of the *file_spec* arguments found in the specified snapshot.

You can also restore snapshots for non-existent *file_spec* arguments. This happens when the value you specify no longer exists on the host, but existed when you took the snapshot. For example, it might be a file system that you have now unmounted or a disk group that you have removed.

Normally, you restore snapshots from the host where you took the snapshots (in other words, the originating host). You can also restore snapshots using a different, or non-originating, host (see "Restoring a snapshot from a different host" on page 145).

How SnapDrive restores snapshots

SnapDrive for UNIX performs the following operations when you restore snapshots:

- When you restore snapshots for disk groups, or for host volumes and file systems that are created on them, SnapDrive for UNIX restores the whole disk group. If you specify part of a disk group, SnapDrive still restores the entire disk group. An error occurs if you enter only a subset of the host volumes and/or file systems in each disk group on the command line. You can include the `-force` option to override this error; however, SnapDrive for UNIX then restores the entire disk group.
- When you restore a snapshot for a file system created directly on a LUN, SnapDrive for UNIX restores the LUN where the file system resides and mounts the file system.
- When you restore snapshots of LUNs (`-lun`), SnapDrive restores the LUNs you specify.

  If you use the `-lun` option and specify a snapshot that contains file system, disk group, or host volume entities, SnapDrive restores the LUN that you specify without restoring the storage entity. You must enter the long name for LUN, and you must use the `-force` option.

- When you restore an NFS directory tree, SnapDrive restores all the directories and files in the directory tree. You can also restore individual NFS files. Within the directory tree, SnapDrive will delete any new NFS files or directories that you create after you created the snapshot.
- If the configuration of the disk group you're trying to restore has changed since the snapshot was taken, the restore operation fails. If you have added or removed a host volume, file system, or LUNs, changed the way your data is striped, or resized any volume manager entity above the disk group level, you can override and restore an older snapshot by including the `-force` option.

  Note
  You should always take a new snapshot whenever a LUN or NFS directory tree has been added to or removed from a disk group.

- You cannot restore snapshots of
  - Root disk groups
  - Shared disk group
  - Boot device
  - Swap device

Restoring snapshots on mirrored filers

When you create a snapshot on a mirrored storage system, the snapshot is automatically repilcated, from the source system where it is created, to the destination (mirrored) storage system. SnapDrive for UNIX allows you to restore the snapshot on the source storage system. You can also restore the snapshot on the destination storage system, provided you meet the following guidelines.

Restoring a single storage entity on a filer or filer cluster: You can restore a snapshot that contains a single storage entity that resides on a storage system or on a clustered storage system. The name of the volume on the destination storage system must match the name of the volume on the source storage system.

Restoring multiple storage entities: To restore a snapshot that contains storage entities that reside on multiple destination storage systems you must meet the following requirements:

- The storage entities you specify on the command line must reside on a single storage system, or on a clustered storage system pair.
- The name of the volume of the source storage system must match the name of the volume of the destination storage system.

- You must set the `snapmirror-dest-multiple-filervolumes-enabled` arguement in the snapdrive.conf file to `on`.

You can use one command to restore storage entities that reside on a single storate system or on a clustered storage system pair.

Example: In the following example, file system 1 (fs1) resides on filer1, and file system 2 (fs2) resides on filer1 and also on filer2, which is the partner storage system. File system 3 (fs3) resides on filer1, partner filer2, and filer3, which is not part of the cluster. An additional file system, fs4, resides entirely on filer4.

The following command creates a snapshot of fs1, fs2, fs3 and fs4.

```
snapdrive snap create -fs /mnt/fs1 /mnt/fs2 /mnt/fs3 /mnt/fs4
-snapname fs_all_snap
```

The next command restores fs1 and fs2 on the destination filer. Both fs1 and fs2 reside on a clustered pair, so you can restore them with one command.

```
snapdrive snap restore -fs /mnt/fs1 /mt/fs2 -snapname
fs_all_snap
```

The following command restores fs4.

```
snapdrive snap restore -fs /mnt/fs4 -snapname fs_all_snap
```

SnapDrive for UNIX can not restore fs3 on the destination storage system, because this file system resides on filer1, filer 2 and filer 3.

Restoring a snapshot

The restore operation can take several minutes, depending on the type and amount of data being restored.

Caution

Do not interrupt the restore operation by entering Ctrl-C. Doing so could leave your system in an unusable configuration. If that happens, you will need to consult with technical support in order to recover.

To restore a snapshot, complete the following steps.

Step	Action
1	Make sure you are not in **any** directory on a file system that you want to restore. You can perform the `snap restore` command from any directory except the one on a file system to which you want to restore the information.

Step	Action
2	Execute the `snapdrive snap restore` command. This command has the following format: `snapdrive snap restore -snapname` *`snap_name`* `{ -lun | -dg | -fs | -hostvol | -file }` *`file_spec`* `[`*`file_spec`* `...] [{ -lun | -dg | -fs | -hostvol | -file }` *`file_spec`* `[`*`file_spec`* `...] ...] [-force [-noprompt]] [{-reserve | -noreserve}]` **Note** For details about using these options and arguments, see the section "SnapDrive for UNIX keywords and arguments" on page 292. ◆ *snap_name* is the name of the snapshot. It can be either a short name, such as mysnap1, or a long name that includes the storage system name, volume, and snapshot name Generally, NetApp recommends that you use the short name, unless the *file_spec* argument is non-existent (i.e., no longer exists on the host, see the explanation of the *file_spec* argument). Then you must use the long name for the snapshot. ◆ By default, SnapDrive for UNIX determines the storage system and storage system volume based on the value you supply for the *file_spec* argument. If you use a long name for the snapshot and the pathname does not match the storage system/storage volume information on the command line, SnapDrive for UNIX fails.

Step	Action
	◆ *file_spec* is the name of the storage entity you specify.You can specify NFS entities, LUNs, file systems created directly on LUNs and LVM entities on the same command line. ❖ If you use the `-dg` option, *file_spec* is the name of that disk group. ❖ If you use the `-fs` option, *file_spec* is the NFS or file system mount point. ❖ If you use the `-file` option, the *file_spec* is the name of the NFS file. This option is only for files that reside in NFS directory trees. ❖ If you use the `-lun` option, *file_spec* is the name of the LUN. If the *file_spec* is a storage entity that uses LUNs (LUNs, file systems on LUNs, or LVM entities), you must include the storage system, volume and LUN name in the `-lun` specification. If the snapshot contains storage entities that span multiple storage system volumes, you can restore any of the entities in that snapshot ❖ In some cases the value supplied with the *file_spec* argument may not exist on the host. For example, if you had unmounted a file system or removed a disk group by exporting it, deporting it, or destroying it, that file system or disk group could still be a value for the *file_spec* argument. It would, however, be considered a non-existent value. SnapDrive for UNIX can restore snapshots for such a non-existent *file_spec*, but you must use the long snapshot name.

Step	Action

	When you restore snapshots that span multiple storage systems and volumes and contain an non-existent *file_spec* argument, SnapDrive for UNIX permits an inconsistency in the command line that it does not allow for existing *file_spec* arguments. If you want to restore only one storage entity from a multiple storage system snapshot, the snapshot you specify does not need to be on the same storage system as the storage system containing the storage entity. For example, a snapshot might contain dg2, which was created on filer2, and dg3, which was created on filer3. To restore dg2 only, you can specify either the snapshot with the pathname to filer2 or the pathname to filer 3. For more information, see "Example of restoring to a non-existent disk group" on page 144.
	◆ If you supply a non-existent *file_spec* argument that, at the time the snapshot was created, spanned multiple storage system volumes, you can specify any of the snapshots in which the values for the *file_spec* argument are stored.
	For example, suppose disk group dg1 spans filer1:/vol/vol1 and filer2:/vol/vol1. When you performed the `snap create` operation `snap create -dg dg1 -snapname snapdg1`, SnapDrive for UNIX would have created two snapshots: filer1:/vol/vol1:snapdg1 and filer2:/vol/vol1:snapdg1
	If you want to restore dg1 and it currently exists on the host (i.e., it is *live*), you only need to supply the short name of the snapshot on the command line and SnapDrive for UNIX finds both snapshots and restores them:
	`snap restore -dg dg1 -snapname snapdg1`
	If you exported disk group dg1, it becomes a.non-existent entity on the host. You can still restore the snapshot, but now you must specify the long snapshot name that contains the storage system and volume information. The `snap restore` command only accepts one snapshot name, so you can supply the name of either snapshot that was created by the `snap create` command. Enter one of the following command lines:
	`snap restore -dg dg1 -snapname filer1:/vol/vol1:snapdg1`
	`snap restore -dg dg1 -snapname filer2:/vol/vol1:snapdg1`

Step	Action
	◆ Use the - `reserve	-noreserve` option to specify whether or not SnapDrive creates a space reservation when you restore the snapshot. ◆ You must include the `-force` option on the command line if you attempt to restore a disk group where the configuration has changed since the last snapshot. For example, if you changed the way data is striped on the disks since you took a snapshot, you would need to include the `-force` option. Without the `-force` option, this operation fails. This option ask you to confirm that you want to continue the operation unless you include the `-noprompt` option with it. **Note** If you added or deleted a LUN, the restore operation fails, even if you include the `-force` option. **Result:** SnapDrive for UNIX replaces the existing contents of the LUNs you specify in the `snap restore` command line with the contents of the LUNs in the snapshot you specify. This operation could take several minutes. When the operation is complete, SnapDrive for UNIX displays a message similar to the following: `Snap restore <filespec list> succeeded.` **Examples of command lines:** The following are examples of the `snapdrive snap restore` command: `snapdrive snap restore -fs /mnt/dir -snapname filer1:/vol/vol1:NewSnap33` `snapdrive snap restore -lun filer1:/vol/vol1/luna lunb -snapname Sunday` `snapdrive snap restore -lun filer1:/vol/vol1/luna filer2:/vol/vol1/lunb -snapname Thursday` `snapdrive snap restore -vg OracleVg1 Datavg -snapname Monday` `snapdrive snap restore -dg dg1 dg2 -snapname Tuesday`

Step	Action

	This example uses a host called DBserver and a disk group dg1 that has host volumes myvol1 and myvol2. The volume dg1/myvol2 has a file system mounted on /fs2. The disk group dg1 has three LUNs in it: toaster:/vol/vol1/lun0, toaster:/vol/vol1/lun1, and toaster: /vol/vol1/lun2. You could use any of the following commands to restore the same data:
	
	`snapdrive snap restore -dg dg1 -snapname snap1`
	
	`snapdrive snap restore -v dg1 -snapname snap1`
	
	`snapdrive snap restore -hostvol dg1/myvol2 dg1/myvol1 -snapname toaster:/vol/vol1:snap1`
	
	`snapdrive snap restore -lvol dg1/myvol1 -fs /fs2`
	
	Example on a Linux host: This example restores a snapshot on a Linux host.
	
	`# snapdrive snap restore -dg dg5 -snapname bagel5snapped`
	`Starting to restore LUNs in disk group dg5`
	`  WARNING:  This can take several minutes.`
	`  DO NOT CONTROL-C!`
	`  If snap restore is interrupted, the disk group being restored may have inconsistent or corrupted data.`
	`  For detailed progress information, see the log file /var/log/sd-recovery.log`
	`  Importing dg5`
	`snap restore: snapshot bagel5snapped contains:`
	`  disk group dg5 containing host volumes`
	`  vol1`
	`  vol2`
	`  vol3`
	`snap restore: restored snapshot surf:/vol/vol1:bagel5snapped`
	`#`

Step	Action

	Example of restoring to a non-existent disk group: This example takes you through the steps of creating a snapshot across multiple storage systems and volumes and then restoring it, even though the disk group it contains no longer exists on the host. The example uses disk group dg2, which is on filer2:/vol/vol2, and disk group dg3, which is on filer3:/vol/vol3. The following `snap create` command was used:
	`snap create -dg dg2 dg3 -snapname snapdg23`
	This command created the snapshots filer2:/vol/vol2:snapdg23 and filer3:/vol/vol3:snapdg23
	If dg2 and dg3 still exist on the host, you can use the short name to restore the snapshot:
	`snap restore -dg dg2 dg3 -snapname snapdg23`
	Suppose, however, that you had exported both dg2 and dg3. In that case, they no longer exist on the host. You can restore them by specifying either of the snapshots created by the `snap create` command line, but you must include the long snapshot name on the snap restore command line.You could enter either of the following command lines:
	`snap restore -dg dg2 dg3 -snapname filer3:/vol/vol3:snapdg23`
	`snap restore -dg dg2 dg3 -snapname filer2:/vol/vol2:snapdg23`
	If you only want to restore dg2, enter the one of the following command lines:
	`snap restore -dg dg2 -snapname filer2:/vol/vol2:snapdg23`
	`snap restore -dg dg2 -snapname filer3:/vol/vol3:snapdg23`
	Because dg2 no longer exists on the host, you can specify the snapshot on filer3 even though dg2 was created on filer2. You can only use this inconsistency if the snapshot spans multiple storage systems or volumes and the host entity no longer exists. If dg2 still existed on the host, the second command line would fail.

Step	Action

	Example of restoring LUNs without restoring storage entities: The following shows how to use the `-lun` option to restore a specific LUN without restoring the storage entities that it contains. This syntax is useful in cases where a LUN that contains a storage entity is accidently removed.
	
	In the following example, disk group 1 (dg1) resides on luna and lunb. The command restores luna, which was removed accidently, but does not restore lunb or dg1. The long lun name and `-force` option are required.
	
	`snapdrive snap restore -lun toaster:/vol/vol1/luna -snapname dg1_snapshot -force`
	
	Example of warning messages: The following is the standard warning message that appears during a `snap restore` operation:
	
	`Starting to restore LUNs in disk groups: <dg1>`
	`WARNING:  This can take several minutes.`
	`DO NOT CONTROL-C!`
	`If snap restore is interrupted, the disk groups being restored may have inconsistent or corrupted data.`
	
	SnapDrive for UNIX also displays a warning message during a restore operation if any logical volumes have been added or removed since the snapshot was taken. The message tells you which logical volumes have been added or removed. For example, if you added a volume called /dev/tester2/vol2 and removed the volume /dev/tester2/vol3 after you took the snapshot snap1, you would see the following message when you executed the `snap restore` command:
	`Warning (Warning 0001-178): diskgroup configuration has changed since snapshot lobster:/vol/vol1:snap1 was taken:`
	`added hostvol /dev/tester2/vol2`
	`removed hostvol /dev/tester2/vol3`

Restoring a snapshot from a different host

In most cases, you restore a snapshot from the host where you took the snapshot. On occasion, you might need to restore a snapshot using a different, or non-originating, host. To restore a snapshot using a non-originating host, use the same `snapdrive snap restore` command that you would normally use. If the

snapshot you restore contains NFS entities, the non-originating host must have permission to access the NFS directory. Refer to “NFS considerations” on page 18 for additional information.

Connecting to a snapshot

Connecting to a snapshot

SnapDrive for UNIX lets you connect to a snapshot from a different location on a host. This new location can be on the host where you took the snapshot (the originating host) or on a different host (the non-originating host).

Being able to set up the snapshots in a new location means you can back up a snapshot to another medium, perform maintenance on a disk group, or simply access the snapshot data without disrupting the original copy of the data.

With this command, you can connect a snapshot that contains:

- LUNs
- A file system created directly on a LUN
- Disk groups, host volumes and file systems created on LUNs
- NFS directory trees

How snap connect works

When you use the `snapdrive snap connect` command, it clones the storage for the entity you specify and imports it to the host.

- If you specify a snapshot that contains a LUN (`-lun`), SnapDrive maps a new copy of the LUN to the host. You can not use the `snapdrive snap connect` command to specify a LUN on the same command line with other storage entities (`-vg`, `-dg`, `-fs`, `-lvol`, or `-hostvol`).
- If you specify a file system that resides directly on a LUN, Snapdrive maps the LUN to the host and mounts the file system.
- If you specify a snapshot that contains a disk group, or a host volume or file system that is part of a disk group, the `snap connect` command connects the entire target disk group. To make the connection, Snapdrive re-activates all of the logical volumes for the target disk group and mounts all the file systems on the logical volumes.
- If you specify a snapshot that contains an NFS directory tree, SnapDrive creates a clone of the FlexVol volume that contains the NFS directory tree. SnapDrive then connects the volume to the host and mounts the NFS file system. Within the directory tree, SnapDrive will delete any new NFS files or directories that you create after you created the snapshot. SnapDrive will delete from the FlexVol volume any files or directories that are outside of the NFS directories that you connect.

If you connect a snapshot that contains NFS directory trees using the `-readonly` option, SnapDrive mounts the snapshot of the directory directly without creating a clone.

You can not use the `snapdrive snap connect` command to specify NFS mount points on the same command line as non-NFS entities (i.e., using the optoins `vg, -dg`, `-fs`, `-lvol`, or `-hostvol`).

Connecting snapshots on mirrored filers

When you create a snapshot on a mirrored storage system, the snapshot is automatically repilcated, from the source system where it is created, to the destination (mirrored) storage system. SnapDrive for UNIX allows you to connect the snapshot on the source storage system. You can also connect the snapshot on the destination storage system, provided you meet the following guidelines.

Connecting a single storage entity on a filer or filer cluster: You can connect a snapshot that contains a single storage entity that resides on a storage system or on a clustered storage system. The name of the volume on the destination storage system must match the name of the volume on the source storage system.

Connecting multiple storage entities: To connect a snapshot that contains storage entities that reside on multiple destination storage systems you must meet the following requirements:

- The storage entities you specify on the command line must reside on a single storage system, or on a clustered storage system pair.
- The name of the volume of the source storage system must match the name of the volume of the destination storage system.
- You must set the `snapmirror-dest-multiple-filervolumes-enabled` arguement in the snapdrive.conf file to `on`.

You can use one command to connect storage entities that reside on a single storate system or on a clustered storage system pair.

Example: In the following example, file system 1 (fs1) resides on filer1, and file system 2 (fs2) resides on filer1 and also on filer2, which is the partner storage system. File system 3 (fs3) resides on filer1, partner filer2, and filer3, which is not part of the cluster. An additional file system, fs4, resides entirely on filer4.

The following command creates a snapshot of fs1, fs2, fs3 and fs4.

```
snapdrive snap create -fs /mnt/fs1 /mnt/fs2 /mnt/fs3 /mnt/fs4
-snapname fs_all_snap
```

The next command connect fs1 and fs2 on the destination filer. Both fs1 and fs2 reside on a clustered pair, so you can restore them with one command.

```
snapdrive snap connect -fs /mnt/fs1 /mt/fs2 -snapname
fs_all_snap
```

The following command restores fs4.

```
snapdrive snap connect -fs /mnt/fs4 -snapname fs_all_snap
```

SnapDrive for UNIX can not connect fs3 on the destination storage system, because this file system resides on filer1, filer 2 and filer 3.

snap connect and snap restore

Unlike the `snap restore` operation, the `snap connect` operation does not replace the existing contents of the LUNs that make up the host entity with the snapshot contents. It clones the information.

Once the connection is made, both `snap connect` and `snap restore` perform similar activities.

- The `snap connect` command activates logical volumes for the storage entity, mounts file systems, and optionally adds an entry to the host file system table.
- The `snap restore` command activates the logical volumes for the storage entity, mounts the file systems, and applies the host file system mount entries that were preserved in the snapshot.

Guidelines for connecting snapshots

Follow these guidelines when connecting snapshots:

- The `snapdrive snap connect` command only works with snapshots created with version 2.x of SnapDrive for UNIX. It does not work with snapshots created using a version 1.x of SnapDrive for UNIX.
- There is a problem performing `snap connect` operations on an AIX host if the following is true:
  - The host is running AIX 5.2.
  - The SANpath version is 4.0.5l and has the SANpath workaround enabled (the f`ailback` attribute for spdctrl is set to 3).

    In this case, the `snap connect` operation fails. The workaround is to disable SANpath's PVID feature. To do this, enter the following command:

    setsp -W0

    SANpath 4.0.5m fixes this issue.

- On Linux hosts, the `snap connect` command is supported providing the snapshot you connect contains a LUN, or a LUN with a file system, that was created without activating the Linux LVM. SnapDrive for UNIX does not support the `snap connect` command for Linux entities that are created using the Linux LVM. Refer to "Connecting to a snapshot" on page 147 for additional information.

  The snapdrive snap connect command does not permit you to rename the disk group on a Linux host. For example, the command `snapdrive snap connect -dg dg1 dg1copy -snapname toaster:/vol/vol1:dg1snapshot` is not supported.
- For read and write access to NFS directory trees, the `snapdrive snap connect` command uses the Data ONTAP FlexVol volume feature, and therefore requires Data ONTAP 7.0. Configurations with Data ONTAP 6.5 can connect NFS files or directory trees, but are provided with read only access.

Connecting snapshots that contain LUNs

To connect a snapshot that contains LUNs, complete the following step.

Step	Action
1	Execute the `snapdrive snap connect` command: `snapdrive snap connect -lun` *`s_lun_name d_lun_name`* `[[-lun]` *`s_lun_name d_lun_name`* `...] -snapname` *`long_snap_name`* `[-igroup ig_name [ig_name ...]]`

Step	Action
	Keep the following in mind when you execute this command: Each *s_lun_name* argument indicates a LUN entity that was captured in the snapshot specified by *long_snap_name*. The *d_lun_name* allows you to specify a destination name that SnapDrive for UNIX uses to make the LUN available in the newly-connected copy of the snapshot. The argument has the following format ◆ The `-lun` argument indicates that a LUN is the storage entity that is captured in the snapshot specified by *s_lun_name*. You can not use the `snapdrive snap connect` command to specify a LUN (`-lun`) on the same command line with other storage entities (`-vg`, `-dg`, `-fs`, `-lvol`, or `-hostvol`). ◆ The *s_lun_name* specifies a LUN that exists in the `-snapname` *`long_snap_name`*. The short *lun_name* is required. You cannot include a storage system or storage system volume name. ◆ The *d_lun_name* specifies the name at which the LUN will be connected. The short *lun_name* is required. You cannot include a storage system or storage system volume name. You must specify a *d_lun_name.* ◆ This command fails if you specify a destination name that is already in use.

Step	Action

	Result: SnapDrive for UNIX clones the LUNs you specify and connects them to a new location.
	
	Example 1: The following example connects the LUN mylun1, in hornet/vol/vol/tuesdaysnapshot to mylun1copy.
	`# ./snapdrv snap connect -lun mylun1 mylun1copy -snapname hornet:/vol/vol1:somesnapshot`
	
	`connecting hornet:/vol/vol1/mylun1:`
	`LUN copy mylun1copy ... created`
	`(original: hornet:/vol/vol1/mylun1)`
	
	`mapping new lun(s) ... done`
	`discovering new lun(s) ... done`
	
	Example 2: The following example connects two LUNs, mylun1 and mylun2, to mylun1copy and mylun2copy, respectively.
	`# ./snapdrv snap connect -lun mylun1 mylun1copy -lun mylun2 mylun2copy -snapname hornet:/vol/vol1:tuesdaysnapshot`
	
	`connecting hornet:/vol/vol1/mylun1:`
	`LUN copy mylun1copy ... created`
	`(original: hornet:/vol/vol1/mylun1)`
	
	`mapping new lun(s) ... done`
	
	`connecting hornet:/vol/vol1/mylun2:`
	`LUN copy mylun2copy ... created`
	`(original: hornet:/vol/vol1/mylun2)`
	
	`mapping new lun(s) ... done`
	`discovering new lun(s) ... done`

Connecting snapshots that contain storage entities

To connect snapshots that contain storage entities other than LUNs, complete the following step.

Step	Action
1	Execute the `snapdrive snap connect` command: `snapdrive snap connect fspec_set [fspec_set...] -snapname long_snap_name [-igroup ig_name [ig_name ...]] [-autoexpand] [-autorename] [-nopersist] [{-reserve | -noreserve}] [-readonly]` **Note** On Linux hosts, the `snap connect` command is not supported on the originating host due to potential disk group ID conflicts. SnapDrive supports this type of connection only when the snapshot contains a LUN, or a file system created directly on a LUN, because these entities do not involve the LVM. For details about using the options and arguments in these command lines, see the "SnapDrive for UNIX keywords and arguments" on page 292.

Step	Action
	Keep the following in mind when you execute this command: ◆ Each *fspec_set* argument identifies a storage entity that was captured in the snapshot specified by *long_snap_name*. The argument also allows you to specify a set of destination names that SnapDrive for UNIX uses to make the entity available in the newly-connected copy of the snapshot. The argument has the following format: `{ -dg | -fs | -hostvol } src_fspec [dest_fspec] [{-destdg | -destvg } dg_name] [{-destlv | -desthv } lv_name]` **Note** The `-destdg` and `-destvg` options and the `-dg` and `-vg` options are synonyms that reflect the fact that some operating systems refer to disk groups and others refer to volume groups. The `-destlv` and `-desthv` options and the `-hostvol` and `-lvol` options are also synonyms. This guide uses `-destdg` to refer to both disk groups and volume groups and `-desthv` to refer to both logical volumes and host volumes. For details about using all the options and arguments available with this command, see "SnapDrive for UNIX keywords and arguments" on page 292. ◆ This command fails if you specify a destination name that is already in use. It also fails if you specify a file system name that is being used as a mount point. ◆ The role of the *src_fspec* argument is to identify an entity within the snapshot named `long_snap_name`. SnapDrive for UNIX uses this name to locate the entity that needs to be connected in order to make the data available. If you connect a snapshot that specifies an NFS mount point, you can not specify non-NFS entities on the same command line (`vg`, `-dg`, `-fs`, `-lvol`, or `-hostvol`).

Step	Action
	◆ The role of the *dest_fspec* argument, if given, is to identify where you're putting the snapshot information. It specifies the name by which the target entity will be accessible after its disk group or LUNs are connected. If you do not supply a name for connecting a destination entity, the `snap connect` command tries to use the name from the source group. This means that if you omit the *dest_fspec* value, `snap connect` uses the value in *src_fspec*. ◆ You can use the *-destxx* options to specify names for destination storage entities if this information is not part of the *dest_fspec*/*src_fspec* pair. For example, the `-fs` option names only a destination mount point so you can use the `-destdg` option to specify the destination disk group. If you do not specify the name needed to connect an entity in the destination disk group, the `snap connect` command takes the name from the source disk group. ◆ The `-autoexpand` and `-autorename` options apply to each disk group specified on the `snap connect` command line and any host LVM entity within that group (i.e., the options apply across every *fspec_set* argument given on the command line). When you supply either of these options on the command line, you must also include either the name of the disk group or some of the LVM entities in it. The `-autoexpand` option simplifies the process of identifying the items in the disk group. You can supply a subset of the logical volumes or file systems in the disk group and SnapDrive for UNIX then expands the connection to the rest of the logical volumes or file systems in the disk group. Without the `-autoexpand` option default, you must specify *all* affected host volumes and file systems contained in that disk group in order to connect the entire disk group. **Note** If the value you enter is a disk group, you don't need to enter all the host volumes or file systems because SnapDrive for UNIX knows what the disk group is connecting to.

Step	Action
	◆ The `-autorename` option tells SnapDrive for UNIX to generate a new, unused name for the destination entity if the default name is in use. ◆ When you use the `-autoexpand` option without the `-autorename` option, the `snap connect` command fails if the default name for the destination copy of an LVM entity is in use. If you include the `-autorename` option, SnapDrive for UNIX renames the entity when the default name is in use. This means that with the `-autorename` option on the command line, the snap connect continues regardless of whether all the necessary names are available. ◆ NetApp strongly recommends that you use the default igroup to contain the LUNs being connected instead of specifying an igroup on the target storage system. ◆ The `-nopersist` option allows you to connect a snapshot to a new location without creating an entry in the host file system table (for example, `fstab` on Linux). By default SnapDrive creates persistent mounts. This means that: ❖ When you connect a snapshot on a Solaris, AIX or HP-UX host, SnapDrive mounts the file system and then places an entry for the LUNs that comprise the filesystem in the host file system table file. ❖ When you connect a snapshot on a Linux host, SnapDrive mounts the file system, resets the file system UUID and label, and places the UUID and mount point in the host file system table file. You can not use `-nopersist` to connect a snapshot that contains an NFS directory tree. ◆ The `-reserve` | -noreserve option allows you to specify whether or not SnapDrive creates a space reservation when you connect the snapshot.
	◆ The `-readonly` option allows you to connect NFS directory trees to Data ONTAP 6.5 and Data ONTAP 7.0 configurations. ❖ If your configuration uses Data ONTAP 6.5, you must specify this option to connect the NFS entities from the snapshot. You must also use this option for later versions of Data ONTAP if the configuration uses traditional (not FlexVol) volumes. SnapDrive for UNIX provides read only access to snapshots on configurations with traditional volumes. ❖ If your configuration uses Data ONTAP 7.0, and also uses FlexVol volumes, SnapDrive automatically provides read and write access to snapshots you connect. Use the `-readonly` option to restrict access to read only.

Step	Action

	Result: SnapDrive for UNIX clones the storage in the entity you specify and imports it to a new location. You can then use the snapshot in that location.
	
	Example 1: The following are command lines you can use with snap connect.
	
	`# snapdrive snap connect -hostvol srcdg/myvol_1 destdg/myvol_2 -snapname toaster:/vol/vol1:mysnapshot`
	
	`# snapdrive snap connect -vg myvg -snapname toaster:/vol/vol1:snap1`
	
	`# snapdrive snap connect -dg src_dg1 dest_dg1 -snapname toaster:/vol/vol1:mysnapshot`
	
	Example 2: The following command line connects a disk group and uses the default names as the destination names (that is, it creates them from the source names).
	
	`# snapdrive snap connect -vg vg1 -snapname filer1:/vol/vol1:vg1snapshot`
	
	`connecting vg1:`
	`LUN copy vg1_lun1_0 ... created`
	`(original: filer1:/vol/vol1/vg1_lun1)`
	
	`mapping new lun(s) ... done`
	`discovering new lun(s) ... done`
	`Importing vg1`

	Example 3: The following command line connects a disk group with a single host volume. It also specifies a name for the destination host volume and disk group.
	
	`# snapdrive snap connect -lvol vg1/vol1 vg1copy/vol1copy -snapname filer1:/vol/vol1:vg1snapshot`
	
	`connecting vg1:`
	`LUN copy vg1_lun1_0 ... created`
	`(original: filer1:/vol/vol1/vg1_lun1)`
	`mapping new lun(s) ... done`
	`discovering new lun(s) ... done`
	`Importing vg1copy`

Step	Action

	Example 4: The following command line connects a disk group with two LUNs and two file systems. It specifies a destination name for each of the file systems, the host volume for one of the file systems, and the disk groups for both file systems.
	
	`# snapdrive snap connect -fs mnt/fs1 /mnt/fs1copy -destvg vg1copy \`
	`-fs /mnt/fs2 /mnt/fs2copy -destlv vg1copy/vol2copy -destvg vg1copy \`
	`-snapname filer1:/vol/vol1:vg1snapshot`
	
	`connecting vg1:`
	`LUN copy vg1_lun1_0 ... created`
	`(original: filer1:/vol/vol1/vg1_lun1)`
	`LUN copy vg1_lun2_0 ... created`
	`(original: filer1:/vol/vol1/vg1_lun2)`
	
	`mapping new lun(s) ... done`
	`discovering new lun(s) ... done`
	`Importing vg1copy`

	Example 5: The following command line includes the `autoexpand` option as it connects a disk group with two file systems. It uses the default names as the destination names (that is, it creates them from the source names).
	
	`# snapdrive snap connect -lvol mnt/fs1 -snapname filer1:/vol/vol1:vg1snapshot \`
	`-autoexpand`
	
	`connecting vg1:`
	`LUN copy vg1_lun1_0 ... created`
	`(original: filer1:/vol/vol1/vg1_lun1)`
	`LUN copy vg1_lun2_0 ... created`
	`(original: filer1:/vol/vol1/vg1_lun2)`
	
	`mapping new lun(s) ... done`
	`discovering new lun(s) ... done`
	`Importing vg1`

Step	Action

	Example 6: The following command line includes the `autorename` option as it connects a disk group with two file systems and two LUNs.
	
	`# snapdrive snap connect -fs mnt/fs1 -snapname filer1:/vol/vol1:vg1snapshot \`
	`-autorename`
	
	`connecting vg1:`
	`LUN copy vg1_lun1_0 ... created`
	`(original: filer1:/vol/vol1/vg1_lun1)`
	`LUN copy vg1_lun2_0 ... created`
	`(original: filer1:/vol/vol1/vg1_lun2)`
	
	`mapping new lun(s) ... done`
	`discovering new lun(s) ... done`
	`Importing vg1_0`

Disconnecting a snapshot

Using the disconnect operation

You use the `snap disconnect` command to remove the mappings for LUNs, or for storage entities and the underlying LUNs, or for NFS directories in the snapshot.

You can use this command to disconnect snapshots that span multiple storage system volumes or multiple storage systems. The storage entities and volumes can reside on the same storage system or different storage systems.

Use this command to disconnect:

- LUNs
- A file system created directly on a LUN
- Disk groups, host volumes and file systems created on LUNs
- NFS directory trees

The disconnect operation does not modify the connected snapshot. However, by default the operation does delete any temporary LUNs or clones created by the corresponding connect operation.

Note

For LUNs, file systems on LUNs, and LVM entities, this command is equivalent to `snapdrive storage delete`.

Guidelines for disconnecting snapshots

When you disconnect a filesystem, SnapDrive for UNIX always removes the mount point.

Linux hosts allow you to attach multiple fileystems to a single mount point. However, SnapDrive for UNIX requires a unique mount point for each file system. The SnapDrive for UNIX `snap disconnect` command willl fail if you use it to disconnect filesystems that are attached to a single mount point.

For NFS entities: Follow these guidelines when disconnecting snapshots that contain NFS entities:

- If you disconnect an NFS directory tree that you connected with read only permission, Snapdrive:
  - Unmounts the file system
  - Removes the mount entry in the file system table file
  - Removes the mount point
- If you disconnect an NFS directory tree that you connected with read-write permission, Snapdrive:
  - Unmounts the file system
  - Removes the mount entry in the file system table file
  - Deletes the NFS directory tree that corresponds to the file system in the FlexVol volume clone
  - Destroys the underlying FlexVol volume clone (if it is empty)
  - Removes the mount point.

Disconnecting snapshots with LUNs and no storage entities

To disconnect a snapshot that contains LUNs that have no storage entities, complete the following step.

Step	Action

1	Execute the `snapdrive snap disconnect` command:
	`snapdrive snap disconnect -lun` *`long_lun_name`* `[`*`lun_name`* `...]`
	Keep the following in mind when you execute this command:
	◆ Use the `-lun` option to specify a LUN that has no storage entities.You can not use the `snapdrive snap disconnect` command to specify a LUN (`-lun`) on the same command line with other storage entities (`vg`, `-dg`, `-fs`, `-lvol`, or `-hostvol`).
	◆ Specify the `long_lun_name`. This includes the name of the storage system, volume and LUN.
	Result: SnapDrive for UNIX removes the mappings for the storage entities specified in the command line.
	Example: This command removes the mappings to luna and lunb on the storage system toaster
	`# snapdrive snap disconnect -lun toaster:/vol/vol1/luna lunb`

Disconnecting snapshots with storage entities

To disconnect a snapshot that contains a disk group, host volume, file system or NFS directory tree, complete the following step.

Step	Action

1	Execute the `snapdrive snap disconnect` command:
	`snapdrive snap disconnect {-dg | -fs | -hostvol } file_spec [file_spec ...]`
	`[-dg | -fs | -hostvol } file_spec [file_spec ...]] [-full]`
	Note
	The `-dg` and `-vg` options are synonyms. They reflect the fact that some operating systems refer to disk groups and others refer to volume groups. The `-lvol` and `-hostvol` options are also synonyms. This guide uses `-dg` to refer to both disk groups and volume groups and `-hostvol` to refer to both logical volumes and host volumes. For details about all the options and arguments in this command line, see the "SnapDrive for UNIX keywords and arguments" on page 292.
	Keep the following in mind when you execute this command:
	◆ Use the `-full` option to tell SnapDrive for UNIX to disconnect the snapshot even if a host-side entity is named that includes other entities; for example, a disk group is named that has one or more host volumes. If you do not include the `-full` option, you can only specify empty host-side entities. For example, you would need to specify a disk group that does not have any host volumes in it, a host volume that does not contain any file systems, or a file system.
	◆ Make sure the *file_spec* parameter specifies the name of the storage entity you designate.
	If you specify an NFS mount point, you can not use the same command line to specify non-NFS entities (`vg`, `-dg`, `-fs`, `-lvol`, or `-hostvol`).
	◆ An error also occurs if you specify a subset of the host volumes and/or file systems contained in each target disk group.
	Result: SnapDrive for UNIX removes the mappings for the storage entities specified in the command line.

Step	Action

	Example 1: This command line removes the mappings to all the LUNs underlying the host volume dg5/myvolume. It removes any temporary LUNs that were created with a `snap connect` operation.
	`# snapdrive snap disconnect -hostvol dg5/myvolume`
	Example 2: This command disconnects the mapping to disk group 1 (dg1) and to the underlying LUN. It also removes any temporary LUNs that were created with the `snap connect` operation.
	`snapdrive snap disconnect -lun toaster:/vol/vol1/luna -dg dg1`
	Example 3: This command line removes the mapping to the file system fs1, and to the LUN that underlies it. It also removes any temporary LUNs that were created with the `snap connect` operation.
	`# snapdrive snap disconnect -fs mnt/fs1`
	Example 4: This command line removes the mappings to for disk groups dg1, dg2, and dg3. It removes any temporary LUNs that might have been created with the snap connect operation.
	`# snapdrive snap disconnect -dg dg1 dg2 dg3`

Deleting snapshots

Command to use to delete snapshots

The `snapdrive snap delete` command removes the snapshots you specify from a storage system. This command does not perform any operations on the host. It only removes the snapshot from a storage system, if you have permission to do so. (If you want to keep the LUNs and mappings, see the information on the snapdrive storage delete command on "Deleting storage from the host and storage system" on page 216.)

Reasons to delete snapshots

You might delete older snapshots for the following reasons:

- To keep fewer stored snapshots than the hard limit of 255 on a storage system volume

  After this limit is reached, attempts to create new snapshots fail.
- To free space on the storage system volume

  Even before the snapshot limit is reached, a snapshot fails if the disk does not have enough reserved space for it.

Guidelines for deleting snapshots

Follow these guidelines when you use the `snapdrive snap delete` command:

- The `snapdrive snap delete` operation fails if any of the snapshots you want to delete are in use or were not created by SnapDrive for UNIX. You can override this behavior by including the `-force` option on the command line.
- If you have a snapshot that spans multiple storage system volumes, you must manually delete the snapshot on each volume.

Deleting a snapshot

To delete a snapshot, complete the following step.

Step	Action
1	Execute the `snapdrive snap delete` command. This command has the following format: `snapdrive snap delete [-snapname] long_snap_name [snap_name ...] [-verbose] [-force [-noprompt]]` *long_snap_name* is the long form for the name of the snapshot and includes the storage system name, volume, and snapshot name. *snap_name* can be either the long form of the name of the snapshot or the short form (just the snapshot name without path information). When you supply the short form of the name, the `delete` operation uses the storage system (and the storage system volume, if not given) last seen in the argument sequence. You can use the `-verbose` option to display a list of what is being deleted. You can also use the wildcard (*) character in snapshot names. The `snap show` operation enables you to use the wildcard character to show all snapshot names that match a certain pattern. The following rules apply to using wildcards in snapshot names: ◆ You can use a wildcard at the end of the name only. You cannot use the wildcard at the beginning or the middle of a snapshot name. ◆ You cannot use the wildcard in the storage system or storage system volume fields of a snapshot name. See the examples that follow for a sample command line that includes a wildcard. **Note** The section "SnapDrive for UNIX keywords and arguments" on page 292 provides details about using the options and arguments in this command line. **Result:** SnapDrive for UNIX deletes the existing contents of the LUNs you specify in the `snap delete` command line and replaces them with the contents of the LUNs in the snapshot you specify.

Step	Action

	Example 1: This example displays a list of what is being deleted:
	
	`# snapdrive snap delete -v filer1:/vol/vol0/snap1 snap2 snap3`
	`snapdrive: deleting`
	`filer1:/vol/vol0/snap1`
	`filer1:/vol/vol0/snap2`
	`filer1:/vol/vol0/snap3`
	
	Example 2: You can also use the wildcard character to specify the snapshot name:
	
	`# snap delete myfiler:/vol/vol1:mysnap* /vol/vol2:hissnap* yoursnap* hersnap`

Provisioning and Managing Storage 6

About this chapter

This chapter provides details about working with storage provisioning. It contains information on using SnapDrive for UNIX to create and connect to storage.

Topics in this chapter

This chapter discusses the following topics:

- "Overview of storage provisioning" on page 170
- "Creating storage" on page 172
- "Displaying information about storage" on page 185
- "Displaying information about devices and LVM entities" on page 192
- "Connecting LUNs and storage entities to the host" on page 198
- "Connecting only the host-side of storage" on page 207
- "Disconnecting only the host-side of storage" on page 211
- "Deleting storage from the host and storage system" on page 216

Overview of storage provisioning

Using storage provisioning with SnapDrive for UNIX

SnapDrive for UNIX provides end-to-end storage management. With it, you can provision storage from a host to a NetApp storage system and manage that storage with or without using the host LVM.

SnapDrive for UNIX lets you perform the following tasks:

- Create storage by creating LUNs, file systems, logical volumes, and disk groups
- Display information about storage
- Connect to storage
- Resize storage
- Disconnect from the storage
- Delete storage

When you use SnapDrive for UNIX to create storage using the `snapdrive storage create` command, it automatically performs all the tasks needed to set up LUNs, including preparing the host, performing discovery mapping, and connecting to each LUN you create. You can use the `snapdrive storage show` command to display information about the NetApp LUNs, disk groups, host volumes, file systems or NFS directory trees that you create.

SnapDrive for UNIX also provides the `snapdrive storage connect` command. You can use this command to map storage to a new location. This command lets you access existing storage from a different host than the one used to create it. The `snapdrive storage connect` command lets you make existing LUNs, file systems, disk groups and logical volumes accessible on a new host. This operation can be useful if you want to back up a storage entity from the new host to another medium.

The `snapdrive storage resize` command lets you increase the size of your storage in the following ways:

- Specifying a target size that you want the host entity to reach.
- Entering a set number of bytes by which you want to increase the storage.

If at some point, you decide you no longer want your storage mapped to its current location, you can use the `snapdrive storage disconnect` command. This command removes the mappings from one or more host locations to the LUNs making up the storage for that location.

You can also delete the storage. If you execute the `snapdrive storage delete` command, SnapDrive for UNIX removes all the host-side entities you specify as well as all their underlying entities and the LUNs associated with them.

Using storage operations across multiple storage system volumes

SnapDrive for UNIX lets you perform many of the storage operations across multiple storage system volumes as long as they don't manipulate the LVM. This enables you to work with lists of LUNs that exist across multiple storage system volumes.

Creating storage

Creating storage with SnapDrive for UNIX

You can use SnapDrive for UNIX to create the following:

- LUNs
- A file system created directly on a LUN
- Disk groups, host volumes and file systems created on LUNs

SnapDrive for UNIX automatically handles all the tasks needed to set up LUNs associated with these entities, including preparing the host, performing discovery mapping, creating the entity, and connecting to the entity you create. You can also specify which LUNs SnapDrive uses to provide storage for the entity you request.

You do not need to create the LUNs and the storage entity at the same time. If you create the LUNs separately, you can create the storage entity later using the existing LUNs.

Creating storage for LVM entities: If you ask SnapDrive for UNIX to create a logical volume or file system using the LVM, SnapDrive automatically creates the required disk group. SnapDrive creates the file system based on the type supported by the host logical volume manager (LVM). These include:

- AIX: JFS2
- HP-UX: VxFS
- Linux: Ext3
- Solaris: VxFS

Creating storage for a file system that resides on a LUN: If you ask Snapdrive for UNIX to create a file system that resides directly on a LUN, SnapDrive creates and maps the LUN, then creates and mounts the file system without involving the host LVM.

Methods for creating storage

To make it easier to create the storage you want, SnapDrive for UNIX provides some basic formats for the `snapdrive storage create` command. This is because the `storage create` operations fall into the following general categories:

- Creating LUNs. This command automatically creates the LUNs on the storage system, but does not create any additional storage entities. SnapDrive for UNIX performs all of the tasks associated with host

preparation and discovery for each LUN, as well as mapping and connecting to it. See “Creating LUNs without host entities” on page 174 for the steps required to do this.

- Creating a file system directly on a LUN and setting up the LUN automatically. SnapDrive for UNIX performs all the actions needed to set up the file system. You do not need to specify any LUNs for it to create. See “Creating a file system on a LUN and setting up the LUN automatically” on page 175 for the steps required to do this.
- Creating a file system directly on a LUN and specifying the LUN you want associated with it. In this case you use the command to specify the file system you want to set up, and the LUN you want to associate with the file system. See “Creating a file system on a LUN and specifying the LUN” on page 177
- Creating an LVM and setting up the LUN automatically. This command lets you create a file system, a logical volume, or a disk group on the host. SnapDrive for UNIX performs all the actions needed to set up the entity, including automatically creating the required disk group and LUN. You do not need to specify any LUN for it to create. See “Creating an LVM entity and setting up the LUN automatically” on page 178 for the steps required to do this.
- Creating and LVM entity on the host and specifying the LUN you want associated with it. In this case, you use the command to specify both the entity you want to set up (file system, logical volume, or disk group) and the LUN you want associated with that entity. See “Creating an LVM entity and specifying the LUN” on page 182 for the steps required to do this.

Guidelines for the storage create command

Follow these guidelines when using the `snapdrive storage create` command:

- You cannot create a disk group, host volume, or file system using LUNs from more than one storage system volume. If you list LUNs from different storage system volumes with the `-lun` option, you cannot include the `-dg`, `-hostvol`, or `-fs` option on the command line.
- The `-nolvm` option creates a file system directly on a LUN without activating the host Logical Volume Manager. You can not specify host volumes or disk groups when you use this option.
- You can not use SnapDrive for UNIX storage provisioning commands for NFS files or directory trees.
- If you use the `snapdrive storage create` command to create a file system directly on a LUN, you can not specify more than one LUN. SnapDrive always creates a new LUN when you use this form of the command.

- Some operating systems, such as Linux and Solaris, have limits on how many LUNs you can create. If your host is running one of these operating systems, you may want to run the `storage config` commands. For more information, see "Preparing hosts for adding LUNs" on page 88.

Creating LUNs without host entities

To provision storage by creating LUNs on the storage system, complete the following step.

Step	Action
1	Execute the following `snapdrive storage create` command: `snapdrive storage create -lun long_lun_name [lun_name ...] -lunsize size [{-reserve | -noreserve}] [-igroup ig_name [ig_name ...]]`
	Keep the following in mind when you execute this command: ◆ Use the `-lun` argument to specify the storage system and volume where you want to create the LUNs. The first time you specify a LUN, use the long form of the name (i.e., the full pathname). ◆ If you want to specify additional LUNs, you can use the short form of the name if they are on the same storage system and volume as the first LUN. Otherwise, use the long form of the name again. ◆ Use the `-lunsize` argument to specify the size of each LUN. The default measure for specifying the size is bytes. ◆ Use the `-reserve` | `-noreserve` option to specify whether or not SnapDrive creates a space reservation when you create the storage. ◆ NetApp strongly recommends that you use the default igroup that SnapDrive for UNIX creates to contain the LUNs being connected instead of specifying an igroup on the target storage system. **Result:** SnapDrive for UNIX creates the LUNs you specify. **Example:** This example creates three LUNs on the storage system acctfiler. Each LUN is 10gb. `snapdrive storage create -lun acctfiler:/vol/vol1/lunA lunB lunC -lunsize 10g`

Creating a file system on a LUN and setting up the LUN automatically

To create a file system directly on a LUN, and have SnapDrive for UNIX automatically create the associated LUN, complete the following step.

Step	Action
1	Execute the following `snapdrive storage create` command: `snapdrive storage create -fs` *`file_spec`* `-nolvm [-fstype` *`type`*`] [-fsopts` *`options`*`] [-mntopts` *`options`*`] [-nopersist] -filervol` *`long_filer_path`* `-lunsize` *`size`* `[-igroup` *`ig_name`* `[`*`ig_name`* `...]] [{ -reserve | -noreserve }]` Keep the following in mind when you execute this command: ◆ The `-fs` argument specifies the mount point of the file system you want to create. ◆ The-`nolvm` argument creates the file system on the LUN without activating the host LVM. This option is required. ◆ When you create a file system, you can specify any type (`-fstype`) that SnapDrive for UNIX supports on the host for that operating system. Current values that you can include with this argument are vxfs (Solaris and HP-UX), jfs2 (AIX), or ext3 (Linux). ◆ By default, SnapDrive for UNIX automatically performs all of the tasks associated with host preparation and discovery for the LUN, as well as mapping and connecting to it. If you create a LUN on a Linux host, Snapdrive for UNIX: ❖ Creates the LUN ❖ Configures the LUN into one partition. ◆ You can use the `-fsopts` option to specify options that you want passed to the host operation that creates the new file system (frequently, this operation is the host command `mkfs`). The argument you supply with this option usually needs to be specified as a quoted string and must contain exactly the text to be passed to the host command.

Step	Action
	◆ You can use the `-mntopts` option to specify options that you want passed to the host mount command (for example, to specify host system logging behavior). The options you specify are stored in the host file system table file. Allowable options depend on the host file system type. The `-mntopts` argument is a file system-type option that is specified using the `mount` command "-o" flag. Do not include the "-o" flag in the `-mntopts` argument. For example, the sequence `-mntopts tmplog` passes the string `-o tmplog` to the `mount` command, and inserts the text "tmplog" on a new command line. ◆ The `-nopersist` option allows you to create a file system without creating an entry in the host file system table (for example, `fstab` on Linux). By default the `storage create` command creates persistent mounts. This means that when you create an LVM storage entity on a Solaris, AIX, HP-UX or Linux host, SnapDrive automatically creates the storage, mounts the file system and then places an entry for the file system in the host file system table. On Linux systems, SnapDrive adds a UUID in the host file system table
	◆ Use the `-filervol` option to specify the storage system and volume where you want the LUNs created. Do not specify the LUN. SnapDrive for UNIX creates the LUN automatically when you use this form of the `snapdrive storage create` command. It uses system defaults to determine the LUN IDs, and the size of each LUN. It bases the names of the associated disk/volume groups on the name of the host volume or file system. ◆ To control the size of the host entity, use the `-lunsize size` option to specify the size in bytes of the underlying LUN. ◆ It is strongly recommended that you use the default igroup that SnapDrive for UNIX creates to contain the LUNs being connected instead of specifying an igroup on the target storage system. ◆ Use the `-reserve | -noreserve` option to specify whether or not SnapDrive creates a space reservation when you create the storage. **Result:** SnapDrive for UNIX creates the file system you specify and creates a LUN for it on the storage system you specify. It performs all of the tasks associated with host preparation and discovery for the LUNs, as well as mapping and connecting the LUNs to the host entity. **Example:** This example creates a 100m file system that is created directly on the LUN. `snapdrive storage create -fs /mnt/acct1 -filervol acctfiler:/vol/vol1 -lunsize 100m -nolvm`

Creating a file system on a LUN and specifying the LUN

To create a file system directly on a LUN, and specify the LUNs that are created as part of it, complete the following step.

Step	Action
1	Execute the following `snapdrive storage create` command: `snapdrive storage create -fs` *`file_spec`* `-nolvm [-fstype` *`type`*`] [-fsopts` *`options`*`] [-mntopts` *`options`*`] [-nopersist] -lun` *`long_lun_name`* `-lunsize` *`size`* `[-igroup` *`ig_name`* `[`*`ig_name`* `...]] [{ -reserve | -noreserve }]` Keep the following in mind when you execute this command: ◆ The `-fs` argument specifies the mount point of the file system you want to create. ◆ The-`nolvm` argument creates the file system on the LUN without activating the host LVM. This argument is required. ◆ By default, SnapDrive for UNIX automatically performs all of the tasks associated with host preparation and discovery for the LUN, as well as mapping and connecting to it. If you create a LUN on a Linux host, Snapdrive for UNIX: ❖ Creates the LUN ❖ Configures the LUN into one partition. ◆ When you create a file system, you can specify any type (`-fstype`) that SnapDrive for UNIX supports on the host operating system. Current values that you can include with this argument are vxfs (Solaris and HP-UX), jfs2 (AIX), or ext3 (Linux). ◆ You can use the `-fsopts` option to specify options that you want passed to the host operation that creates the new file system (frequently, this operation is the host command `mkfs`). The argument you supply with this option usually needs to be specified as a quoted string and must contain exactly the text to be passed to the host command.

Step	Action
	◆ You can use the `-mntopts` option to specify options that you want passed to the host mount command (for example, to specify host system logging behavior). The options you specify are stored in the host file system table file. Allowable options depend on the host file system type. The `-mntopts` argument is a file system-type option that is specified using the `mount` command "-o" flag. Do not include the "-o" flag in the `-mntopts` argument. For example, the command `-mntopts tmplog` passes `o tmplog` to the `mount` command, and inserts the text "tmplog" in the host file system table file. ◆ The `-nopersist` option allows you to create a file system without creating an entry in the host file system table file (for example, `fstab` on Linux). By default the `storage create` command creates persistent mounts. This means that when you create an LVM storage entity on a Solaris, AIX, HP-UX or Linux host, SnapDrive automatically creates the storage, mounts the file system and then places an entry for the file system in the host file file system table file. On Linux systems, SnapDrive adds a UUID in the host file system table file.
	◆ Use the `-lun` *`long_lun_name`* option to specify the LUN you want to create. ◆ To control the size of the host entity, use the `-lunsize size` option to specify the size in bytes of the underlying LUN. ◆ It is strongly recommended that you use the default igroup that SnapDrive for UNIX creates to contain the LUNs being connected instead of specifying an igroup on the target storage system. ◆ Use the `-reserve` | `-noreserve` option to specify whether or not SnapDrive creates a space reservation when you create the storage. **Result:** SnapDrive for UNIX creates the file system on the storage system, volume and LUN you specify. It performs all of the tasks associated with host preparation and discovery for the LUNs, as well as mapping and connecting the LUNs to the host entity. **Example:** This example creates a 100m file system on luna, in acctfiler:/vol/vol1. `snapdrive storage create -fs /mnt/acct1 -lun acctfiler:/vol/vol1/luna -lunsize 100m -nolvm`

Creating an LVM entity and setting up the LUN automatically

To create an entity on the host such as a file system, logical volume or disk group and have SnapDrive for UNIX automatically create the associated LUN, complete the following step.

Step	Action
1	Execute the following `snapdrive storage create` command: `snapdrive storage create` *`host_lvm_fspec`* `-filervol` *`long_filer_path`* `-dgsize` *`size`* `[-igroup` *`ig_name`* `[`*`ig_name`* `...]] [{ -reserve | -noreserve }]`

	Keep the following in mind when you execute this command:
	◆ The `host_lvm_fspec` argument lets you specify whether you want to create a file system, logical volume, or disk group. This argument has three general formats. The format you use depends on the entity you want to create.
	To create a file system, use this format:
	`-fs` *`file_spec`* `[-fstype` *`type`*`] [-fsopts` *`options`*`] [-mntopts` *`options`*`] [-nopersist] [ -hostvol` *`file_spec`*`] [ -dg` *`dg_name`*`]`
	To create a logical or host volume, use this format:
	`[-hostvol` *`file_spec`*`] [-dg` *`dg_name`*`]`
	To create a disk or volume group, use this format:
	`-dg` *`dg_name`*
	Note
	The `-dg` and `-vg` options are synonyms that reflect the fact that some operating systems refer to disk groups and others refer to volume groups. In addition, `-lvol` and `-hostvol` are also synonyms. This guide uses `-dg` to refer to both disk groups and volume groups and `-hostvol` to refer to both logical volumes and host volumes. For details about using all the options and arguments available with this command, see "SnapDrive for UNIX keywords and arguments" on page 292.

Step	Action
	If you create a file system, you may also include the host volume specifications, the disk group specifications, or both specifications to indicate the host volume and/or disk group on which the file system will be based. If you do not include these specifications, SnapDrive for UNIX automatically generates the names for the host volume and/or disk group. ◆ When you specify a host volume, SnapDrive for UNIX creates a concatenated host volume. While this is the only format SnapDrive for UNIX supports when creating host volumes, it does allow you to manipulate existing striped host volumes. ◆ When you create a file system, you can specify any type (`-fstype`) that SnapDrive for UNIX supports on the host LVM for that operating system. Current values that you can include with this argument are vxfs (Solaris and HP-UX), jfs2 (AIX), or ext3 (Linux). ◆ You can use the `-fsopts` option to specify options that you want passed to the host operation that creates the new file system (frequently, this operation is the host command `mkfs`). The argument you supply with this option usually needs to be specified as a quoted string and must contain exactly the text to be passed to the host command. ◆ If you are creating an entity such as a file system, you do not need to supply a value for a disk or volume group. SnapDrive for UNIX automatically creates one. ◆ Use the `-filervol` option to specify the storage system and volume where you want the LUN created. ◆ Do not specify any LUNs. SnapDrive for UNIX creates them automatically when you use this form of the `snapdrive storage create` command. It uses system defaults to determine the number of LUNs, the LUN IDs, and the size of each LUN. It bases the names of the associated disk/volume groups on the name of the host volume or file system. ◆ To control the size of the host entity, use the `-dgsize` option to specify the size in bytes of the underlying disk group. ◆ You can use the `-mntopts` option to specify options that you want passed to the host mount command (for example, to specify host system logging behavior). The options you specify are stored in the host file system table file. Allowable options depend on the host file system type. The `-mntopts` argument is a file system-type option that is specified using the `mount` command "-o" flag. Do not include the "-o" flag in the `-mntopts` argument. For example, the command `-mntopts tmplog` passes `-o tmplog` to the mount command line, and inserts the text "tmplog" in the host file system table file.

Step	Action

	◆ The -nopersist option allows you to create a file system without creating an entry in the host file system table file (for example, fstab on Linux). By default the storage create command creates persistent mounts. This means that when you create an LVM storage entity on a Solaris, AIX, HP-UX or Linux host, SnapDrive automatically creates the storage, mounts the file system and then places an entry for the file system in the host file file system table file.
	◆ Use the -reserve | -noreserve option to specify whether or not SnapDrive creates a space reservation when you create the storage.
	
	It is strongly recommended that you use the default igroup that SnapDrive for UNIX creates to contain the LUNs being connected instead of specifying an igroup on the target storage system
	
	Result: SnapDrive for UNIX creates the host entity you specify and creates LUNs for it on the storage system you specify. It performs all of the tasks associated with host preparation and discovery for each of the LUNs, as well as mapping and connecting the LUNs to the host entity.
	
	Example 1: This example creates the file system acctfs with a Solaris file type of VxFS. It sets up LUNs on the storage system acctfiler and creates a disk group that is 1gb.
	
	`snapdrive storage create -fs /mnt/acctfs -fstype vxfs -filervol acctfiler:/vol/acct -dgsize 1g`
	
	Example 2: This example on an HP-UX host creates the file system acctfs and specifies only the storage system volume. SnapDrive for UNIX internally creates and names the entities under the file system.
	
	`snapdrive storage create -fs /mnt/acctfs -filervol acctfiler:/vol/vol1 -vgsize 10g`
	
	`LUN acctfs_SdLun ... created`
	
	`mapping new lun(s) ... done`
	`discovering new lun(s) ... done`
	
	`LUN to device file mappings:`
	`- acctfiler:/vol/vol0/acctfs_SdLun => /dev/rdsk/c9t0d6`
	
	`disk group acctfs_SdDg created`
	`host volume acctfs_SdHv created`
	`file system /mnt/acctfs created`

Creating an LVM entity and specifying the LUN

If you want to both create a host entity such as a file system, logical volume, or disk group and specify the LUN that is created as part of it, complete the following step.

Step	Action

1	Execute the following `snapdrive storage create` command:
	`snapdrive storage create` *`host_lvm_fspec`* `-lun` *`long_lun_name`* `[`*`lun_name`* `...] -lunsize` *`size`* `[-igroup` *`ig_name`* `[`*`ig_name`* `...]] [{ -reserve | -noreserve }]`
	Keep the following in mind when you execute this command:
	Keep the following in mind when you execute this command:
	◆ The `host_lvm_fspec` argument lets you specify whether you want to create a file system, logical volume, or disk group. This argument has three general formats. The format you use depends on the entity you want to create.
	To create a file system, use this format:
	`-fs` *`file_spec`* `[-fstype` *`type`*`] [-fsopts` *`options`*`] [-mntopts` *`options`*`] [-nopersist] [{ -lvol | -hostvol }` *`file_spec`*`] [{ -dg | -vg } dg_name]`
	To create a host volume, use this format:
	`[-hostvol` *`file_spec`*`] [-dg` *`dg_name`*`]`
	To create a disk/volume group, use this format:
	`-dg` *`dg_name`*
	Note
	The `-dg` and `-vg` options are synonyms that reflect the fact that some operating systems refer to disk groups and others refer to volume groups. In addition, `-lvol` and `-hostvol` are also synonyms. This guide uses `-dg` to refer to both disk groups and volume groups and `-hostvol` to refer to both logical volumes and host volumes. For details about using all the options and arguments available with this command, see "SnapDrive for UNIX keywords and arguments" on page 292.
	◆ When you create a file system, you can specify any type (`-fstype`) that SnapDrive for UNIX supports on the host LVM for that operating system. Current values that you can include with this argument are vxfs (Solaris and HP-UX), jfs2 (AIX), or ext3 (Linux).

Step	Action
	◆ You can use the `-fsopts` option to specify options that you want passed to the host operation that creates the new file system (frequently, this operation is the host command `mkfs`). The argument you supply with this option usually needs to be specified as a quoted string and must contain exactly the text to be passed to the host command. ◆ You can use the `-mntopts` option to specify options that you want passed to the host mount command (for example, to specify host system logging behavior). The option you specify is stored in the host file system table file. Allowable options depend on the host file system type. The `-mntopts` argument is a file system-type option that is specified using the `mount` command "-o" flag. Do not include the "-o" flag in the `-mntopts` argument. For example, the command `-mntopts tmplog` passes `-o tmplog` to the mount command line, and inserts the text "tmplog" in the host file system table file. ◆ The `-nopersist` option allows you to create a file system without creating an entry in the host file system table (for example, `fstab` on Linux). By default the `storage create` command creates persistent mounts. This means that when you create an LVM storage entity on a Solaris, AIX, HP-UX or Linux host, SnapDrive automatically creates the storage, mounts the file system and then places an entry for the file system in the host file file system table. On Linux systems, SnapDrive adds a UUID in the host file system table ◆ SnapDrive for UNIX creates a disk/volume group to hold the LUNs based on the value you enter with the `-dg` option. The name you supply for the group must not exist. ◆ Use the `-lun` argument to specify the LUNs you want to create. ◆ Use the `-lunsize` argument to specify the size in bytes of each LUN. The size of the LVM entity depends on the aggregated size of the LUNs you request. ◆ The `-nopersist` option allows you to create a file system without creating an entry in the host file system table (for example, `fstab` on Linux). By default the `storage create` command creates persistent mounts. This means that when you create an LVM storage entity on a Solaris, AIX, HP-UX or Linux host, SnapDrive automatically creates the storage, mounts the file system and then places an entry for the file system in the host file file system table. ◆ It is strongly recommended that you use the default igroup that SnapDrive for UNIX creates to contain the LUNs being connected instead of specifying an igroup on the target storage system. ◆ Use the `-reserve | -noreserve` option to specify whether or not SnapDrive creates a space reservation when you create the storage.

Step	Action

	RESULT: SnapDrive for UNIX creates the host entity and the LUNs you specify.
	
	Example 1: This example creates the file system /mnt/acctfs with an AIX file system type of jfs2. It sets up three LUNs on the storage system acctfiler. Each LUN is 10gb.
	`# snapdrive storage create -fs /mnt/acctfs -fstype jfs2 -lun acctfiler:/vol/vol1/lunA lunB lunC -lunsize 10g`
	`LUN acctfiler:/vol/vol1/lunA ... created`
	`LUN acctfiler:/vol/vol1/lunB ... created`
	`LUN acctfiler:/vol/vol1/lunC ... created`
	
	`mapping new lun(s) ... done`
	`discovering new lun(s) ... done`
	
	`LUN to device file mappings:`
	`- acctfiler:/vol/vol1/lunA => hdisk2`
	`- acctfiler:/vol/vol1/lunB => hdisk3`
	`- acctfiler:/vol/vol1/lunC => hdisk4`
	
	`disk group acctfs_SdDg created`
	`host volume acctfs_SdHv created`
	`file system /mnt/acctfs created`

	Example 2: This example on a Solaris host creates the file system acctfs on three LUNs and explicitly names the volume group and host volume underneath it. Each LUN is 10gb.
	
	`# snapdrive storage create -fs /mnt/acctfs -hostvol acctfsdg/acctfshv -lun acctfiler:/vol/vol1/lunA lunB lunC -lunsize 10g`
	
	`LUN acctfiler:/vol/vol1/lunA ... created`
	`LUN acctfiler:/vol/vol1/lunB ... created`
	`LUN acctfiler:/vol/vol1/lunC ... created`
	
	`mapping new lun(s) ... done`
	`discovering new lun(s) ... done`
	
	`LUN to device file mappings:`
	`- acctfiler:/vol/vol1/lunA => /dev/vx/rdmp/c4t0d3s2`
	`- acctfiler:/vol/vol1/lunB => /dev/vx/rdmp/c4t0d7s2`
	`- acctfiler:/vol/vol1/lunC => /dev/vx/rdmp/c4t0d8s2`
	
	`disk group acctfsvg created`
	`host volume acctfshv created`
	`file system /mnt/acctfs created`

Displaying information about storage

Command to use to display available storage

The `snapdrive storage show` (or `list`) command shows LUNs or NFS directory trees underlying one or more storage entities.You can use the `snapdrive storage show` command to find out what will be in a snapshot of a disk group, host volume, file system or NFS directory tree.

With this command, you can display the following:

- LUNs available for specific storage systems or storage system volumes
- LUNs associated with the file systems, host volumes disk groups
- NFS mount points and directory trees
- LUNs known to a specific host, along with any LVM entities that they include
- Devices known to a specific host

Methods for displaying storage information

To make it easier to display information about storage, SnapDrive for UNIX provides several formats for the `snapdrive storage show` command. This is because storage show operations fall into the following general categories:

- Displaying information about a specific LUN. See "Displaying information about LUNs" on page 186 for the steps to do this.
- Listing information about LUNs available for specific storage systems or storage system volumes. See "Displaying LUNs based on the storage system" on page 187 for the steps required to do this.
- Displaying information about LUNs associated with the arguments you specify. These arguments can include NFS entities, file systems, host volumes, or disk groups. If you use the -verbose option on the command line, SnapDrive for UNIX provides detailed output, such as showing the storage hierarchy including the backing LUNs. See "Displaying information about a storage entity" on page 188 for the steps required to do this.
- Displaying information about the devices known to the host. See "Displaying information about devices" on page 191 for the steps required to do this.
- Displaying information about all devices and LVM entities known to the host. See "Displaying information about devices and LVM entities" on page 192 for the steps required to do this.

Guidelines for the storage show command

The `snapdrive storage show` command has the following guidelines:

- The adapter name is not displayed.
- The `snapdrive storage show -filer` and `snapdrive storage show -filervol` commands do not display output for NFS entities. You must use the `-fs` option to display NFS information.
- The status of non-primary paths is displayed as a hyphen (-) instead of secondary (S).
- If there is no volume manager or host volume underlying a file system, SnapDrive for UNIX displays the following:

```
# snapdrive storage show -fs /pub
 fs: /dev/rdsk/c1t0d1s2         mountpoint: /pub
 device filename adapter path size   state    lun path
 ----------------------------------------------------------
 /dev/rdsk/c1t0d1s2   -   P    10g   online  myfiler:/vol0/lun-1
 /dev/rdsk/c2t0d1s2   -        10g   online  myfiler:/vol0/lun-1
```

- The `snapdrive storage show` command requires time to gather information, and may take a few minutes to display the information you request.

Displaying information about LUNs

To display information about LUNs, complete the following step.

Step	Action

1	Execute the `snapdrive storage show|list` command. To display information about a LUN, use the following format:
	`snapdrive storage { show | list } -lun long_lun_name [lun_name ...] [-verbose] [-quiet]`
	Note
	You can use either `snapdrive storage show` or `snapdrive storage list` in the command line. These commands are synonyms. For details about using all the options and arguments available with this command, see "SnapDrive for UNIX keywords and arguments" on page 292.
	Result: Depending on the command options you select, SnapDrive for UNIX either displays basic information about the LUNs or lists the LUNs associated with the host entity.
	Examples: The following are examples of possible command lines.
	`# snapdrive storage show -verbose -lun BigFiler3:/vol/vol1:lunA lunB lunC`

Displaying LUNs based on the storage system

To display information about the available LUNs on a certain storage system, complete the following step.

Step	Action

1	Execute the `snapdrive storage show` command. This command has the following formats:
	`snapdrive storage { show | list } -filer filer_name [filer_name...] [-verbose] [-quiet]`
	`snapdrive storage { show | list } -filervol long_filer_path [filer_path ...] [-verbose] [-quiet]`
	Note
	You can use either `snapdrive storage show` or `snapdrive storage list` in the command line. These commands are synonyms. For details about using all the options and arguments available with this command, see "SnapDrive for UNIX keywords and arguments" on page 292.
	Keep the following in mind when you execute this command:
	◆ If you want to find out about LUNs on the storage system, use the -filer option. Use the -filervol option if you want to learn about the LUNs available on the storage system volume.
	Result: Depending on the command options you select, SnapDrive for UNIX either displays basic information about the LUNs, or lists the LUNs associated with the host entity.
	Examples: The following examples of possible command lines.
	`# snapdrive storage show -verbose -filer BigFiler3 BigFiler4`
	`# snapdrive storage show -verbose -filervol BigFiler3:/vol/vol1`

Displaying information about a storage entity

To display information about a storage entity, such as an NFS directory tree, file system, disk group, volume group, or host volume, complete the following step.

Step	Action
1	Execute the `snapdrive storage show` command: `snapdrive storage {show | list} {-dg | -fs | -hostvol } file_spec [file_spec ...] [{-dg | -fs | -hostvol } file_spec [file_spec ...]] [-verbose]` **Note** You can use either `snapdrive storage show` or `snapdrive storage list` in the command line. These commands are synonyms. For details about using all the options and arguments available with this command, see "SnapDrive for UNIX keywords and arguments" on page 292. Keep the following in mind when you execute this command: ◆ This form of the command provides information about the specified disk groups, file systems, NFS directory trees, or host volumes, so you must supply a value for either `-dg`, `-fs`, or `-hostvol`. SnapDrive for UNIX looks for this entity on the current host. **Note** The `-dg` and `-vg` options are synonyms that reflect the fact that some operating systems refer to disk groups and others refer to volume groups. In addition, `-lvol` and `-hostvol` are also synonyms. This guide uses `-dg` to refer to both disk groups and volume groups and `-hostvol` to refer to both logical volumes and host volumes. For details about using all the options and arguments available with this command, see "SnapDrive for UNIX keywords and arguments" on page 292. **Result:** SnapDrive displays information about the storage entity you requested.

Step	Action

	Example 1: These are examples of command lines that use the `storage show` command.
	
	`# snapdrive storage list -dg dg1 dg2`
	`# snapdrive storage show -quiet -dg QA_dg_test`
	`# snapdrive storage show -fs /mnt/fs21 /mnt/fs41`
	`# snapdrive storage show -hostvol dg2/vol1 dg4/myvol3`
	`# snapdrive storage list -dg dg2 -fs /mnt/fs42`
	`# snapdrive storage show -hostvol dg2/vol3 -fs /mnt/fs42`
	
	Example 2: This example shows output that appears when you display information about the disk group dg1 on a Solaris host (the output varies slightly depending on which host you use):
	`snapdrive storage show -dg dg1`
	`dg: dg1`
	`hostvol: /dev/vx/dsk/dg1/vol1 state: AVAIL`
	`hostvol: /dev/vx/dsk/dg1/vol3 state: AVAIL`
	`hostvol: /dev/vx/dsk/dg1/vol4 state: AVAIL`
	`fs: /dev/vx/dsk/dg1/vol1 mount point: /pub (persistent)`
	`fs: /dev/vx/dsk/dg1/vol3 mount point: NOT MOUNTED`
	`fs: /dev/vx/dsk/dg1/vol4 mount point: NOT MOUNTED`
	
	`device filename adapter path size proto state lun path`
	`----------- ------- ---- ----- ---- ----- ---------------`
	`/dev/vx/rdmp/c1t0d2s2 - P 10g fcp online myfiler:/vol/vol1/lun2`
	`/dev/vx/rdmp/c1t0d7s2 - P 10g fcp online myfiler:/vol/vol1/lun7`

Step	Action

	Example 3: This example uses an HP-UX host. In this example, SnapDrive for UNIX displays information about the volume group vg01.
	
	`# snapdrive storage show -vg vg01`
	
	`dg: vg01`
	`hostvol: /dev/vg01/lvol1 state: AVAIL`
	`fs: /dev/vg01/lvol1 mount point: NOT MOUNTED`
	
	`device filename adapter path size proto state lun path`
	`---------- ------- ---- ----- ---- ----- ----------`
	`/dev/rdsk/c22t0d0 - P10g 10g fcp online toaster:/vol/vol1/myhp7-1`
	`/dev/rdsk/c20t0d2 - P5g 10g fcp online toaster:/vol/vol1/myhp7-3`
	
	Example 4: This example uses a Solaris host. In this example, SnapDrive for UNIX displays information about the disk group dg1.
	
	`# snapdrive storage show -vg dg1`
	`saving name dg1`
	
	`dg: dg1`
	`hostvol: /dev/vx/dsk/dg1/vol1 state: AVAIL`
	`fs: /dev/vx/dsk/dg1/vol1 mount point: NOT MOUNTED`
	
	`device filename adapter path size proto state lun path`
	`---------- ------- ---- ----- ------ ----- ----------`
	`/dev/vx/rdmp/c1t0d0s2 - P 500m fcp online sky:/vol/vol1/mylun_lun0`

Step	Action

	Example 5: This example uses an AIX host. In this example, SnapDrive for UNIX displays information about the disk group dg1.
	
	`# snapdrive storage show -dg dg1`
	
	`dg: dg1`
	`hostvol: /dev/dg1_log   state: AVAIL`
	`hostvol: /dev/dg1_vol1  state: AVAIL`
	`fs: /dev/dg1_log        mount point: MOUNTED (persistent)`
	`fs: /dev/dg1_vol1       mount point: MOUNTED (persistent)`
	
	`device filename adapter path size proto state lun path`
	`---------- ---- ---- ---- ----- ----- ----------`
	`/dev/rhdisk2 - P 2g fcp online flip:/vol/vol1/nate-aix5_lun0`
	
	`/dev/rhdisk3 - P 2g fcp online flip:/vol/vol1/nate-aix5_lun1:`
	
	Example 6: This example shows the output for an NFS directory tree.
	`# snapdrive storage show -fs /mnt/acctfs_nfs`
	`NFS device: myfiler:/vol/vol1 mount point: /mnt/acctfs_nfs (non-persistent)`

Displaying information about devices

To display information about the LUNs and devices visible to the host, complete the following step.

Step	Action

1	Execute the `snapdrive storage show` command. This command has the following format:
	`snapdrive storage { show | list } -devices`
	Note
	You can use either `snapdrive storage show` or `snapdrive storage list` in the command line. These commands are synonyms. For details about using all the options and arguments available with this command, see "SnapDrive for UNIX keywords and arguments" on page 292.
	Result:: SnapDrive for UNIX displays information about the devices that are visible to the host.
	Examples: The following shows the output SnapDrive displays when you use the `-dev` option.
	`# snapdrive storage show -devices`
	`Connected LUNs and devices:`
	`device filename adapter path size proto state lun path`
	`--- --- --- --- --- --- ---`
	`/dev/vx/dmp/c1t0d3s2 - P 500m fcp online albacore:/vol/vol1/cs_connect_lun1`
	`/dev/vx/dmp/c1t0d4s2 - P 500m fcp online albacore:/vol/vol1/cs_connect_lun2`
	`/dev/vx/dmp/c1t0d5s2 - P 100m fcp online albacore:/vol/vol1/luntestvit1`
	`/dev/vx/dmp/c1t0d0s2 - P 1g fcp online albacore:/vol/vol1/bosun0`
	`/dev/vx/dmp/c1t0d1s2 - P 1g fcp online albacore:/vol/vol1/bosun1`
	`/dev/vx/dmp/c1t0d2s2 - P 1g fcp online albacore:/vol/vol1/bosun2`

Displaying information about devices and LVM entities

To display information about the devices and the LVM entities that are available to the host, complete the following step.

Step	Action
1	Execute the `snapdrive storage show` command. This command has the following format: `snapdrive storage { show | list } -all` **Note** You can use either `snapdrive storage show` or `snapdrive storage list` in the command line. These commands are synonyms. For details about using all the options and arguments available with this command, see "SnapDrive for UNIX keywords and arguments" on page 292. **Result::** SnapDrive for UNIX displays information about the LVM entities and devices that are visible to the host.

Examples: The following shows a portion of the output SnapDrive displays when you use the `-all` option.

```
# snapdrive storage show -all
Connected LUNs and devices:

device filename      adapter  path  size  proto  state  lun path
---------------      -------  ----  ----  -----  ----   --------
/dev/vx/dmp/c1t0d3s2   -       P    500m   fcp   online albacore:/vol/vol1/
cs_connect_lun1
/dev/vx/dmp/c1t0d4s2   -       P    500m   fcp   online albacore:/vol/vol1/
cs_connect_lun2
/dev/vx/dmp/c1t0d5s2   -       P    100m   fcp   online albacore:/vol/vol1/
luntestvit1

Host devices and file systems:

dg: csdg1
hostvol: /dev/vx/dsk/csdg1/hv1_1 state: AVAIL
hostvol: /dev/vx/dsk/csdg1/hv1_2 state: AVAIL
hostvol: /dev/vx/dsk/csdg1/hv1_3 state: AVAIL
fs: /dev/vx/dsk/csdg1/hv1_1 mount point: /mnt/check_submit/csdg1/hv1_1
(persistent)
fs: /dev/vx/dsk/csdg1/hv1_2 mount point: NOT MOUNTED fs: /dev/vx/dsk/csdg1/hv1_3
mount point: NOT MOUNTED
```

Increasing the size of the storage

Resizing the storage to make it larger

SnapDrive for UNIX lets you increase the size of the storage system volume group or disk group. You use the `snapdrive storage resize` command to do this.

Note

This command does not let you resize host volumes or file systems. For example, you can not use the resize command to change the size of a file system on a LUN.

You need to use the LVM commands to resize host volumes and file systems after you have resized the underlying disk group. For information on performing those tasks, see "Resizing host volumes and file systems" on page 197.

You can put the storage resize operations into the following general categories:

- Setting a target size in bytes to which you want to increase the storage.
- Specifying a number of bytes by which you want to increase the storage.

SnapDrive for UNIX adds a system-generated LUN. If you specify an amount by which you want to increase the storage, such as 50MB, it makes the LUN 50MB. If you specify a target size for the storage, it calculates the difference between the current size and the target size. The difference becomes the size of the LUN it then creates.

Guidelines for the storage resize command

Follow these guidelines when you use the SnapDrive storage resize command:

- The storage resize operation can only increase the size of storage. You cannot use it to decrease the size of an entity.
- All LUNs must reside in the same storage system volume.
- The resize operation is not supported directly on logical host volumes, or on file systems that reside on logical hosts volumes or on LUNs. In those cases, you must use the LVM commands to resize the storage.
- You cannot resize a LUN; you must use the `-addlun` option to add a new LUN.

Increasing the size of the storage

To increase the size of your storage, complete the following step.

Step	Action

1	Execute the `snapdrive storage resize` command:
	`snapdrive storage resize -dg file_spec { -growby | -growto } size [-addlun [-igroup ig_name [ig_name ...]]] [{ -reserve | -noreserve}]]`
	Note
	The `-dg` and `-vg` options are synonyms that reflect the fact that some operating systems refer to disk groups and others refer to volume groups. This guide uses `-dg` to refer to both disk groups and volume groups. For details about using all the options and arguments available with this command, see "SnapDrive for UNIX keywords and arguments" on page 292.
	Keep the following in mind as you use this command:
	◆ You must use the `-addlun` option to increase the size of the disk group. This option adds a new, internally-generated LUN to the disk group.
	◆ Use the `-growby` option to add *size* bytes to the entity.
	◆ Use the `-growto` option to increase the size of the entity by the value specified in the *size* argument. You can enter this value as bytes, kilobytes, megabytes, or terabytes.
	◆ Use the `-reserve` | `-noreserve` option to grow storage with or without specifying a space reservation.
	◆ NetApp strongly recommends that you use the default igroup that SnapDrive for UNIX creates to contain the LUNs being connected instead of specifying an igroup on the target storage system.
	Result: This command increases the size of the *file_spec* entity (logical volume or disk group) either by the number of bytes you specify or to the number of bytes you specify.
	◆ Adding bytes to storage (`-growby`)
	◆ Increasing it to the byte size you specify (`-growto`).
	Example 1: This command line increases the size of the disk group my_dg by having it increase to 155mb. In this case, SnapDrive for UNIX automatically increases the size of the last LUN in the disk group until the accumulated disk group size is 155mb.
	`# snapdrive storage resize -dg my_dg -growto 155m`
	`discovering filer LUNs in disk group my_dg...done`
	`resizing LUN x-parrot:/vol/vol0/my_dg_1_SdLun...done`
	`adapting disk group my_dg to lun resize ...done`
	`Disk group my_dg has been resized`

Step	Action

	Example 2: This command line is the same one as in example 1, except that it uses the `-addlun` option to increase the size of the disk group my_dg to 155mb. When you supply the `-addlun` option, SnapDrive for UNIX automatically creates a new LUN in the disk group. It makes the LUN large enough so that the accumulated disk group size reaches 155mb. You do not need to specify any values for the `-addlun` option; SnapDrive for UNIX sets it up automatically.
	
	`# snapdrive storage resize -dg my_dg -addlun -growto 155m`
	`discovering filer LUNs in disk group my_dg...done`
	`LUN x-parrot:/vol/vol0/my_dg_2_SdLun ... created`
	`mapping new lun(s) ... done`
	`discovering new lun(s) ... done.`
	`mapping LUN(s)...done.`
	`initializing LUN(s) and adding to disk group my_dg...done`
	`Disk group my_dg has been resized`
	`#`
	
	Example 3: This command line uses the `-growby` option and increases the size of the disk group testdg to 100m.
	
	`# snapdrive storage resize -dg testdg -growby 100m`
	`discovering filer LUNs in disk group testdg...done`
	`resizing LUN toaster:/vol/vol0/testdg_SdLun...done`
	`adapting disk group testdg to lun resize ...done`
	`Disk group testdg has been resized`
	`#`
	
	Example 4: This command line uses the `-growby` option with the `-addlun` option. SnapDrive for UNIX increases the size of the storage by adding a LUN that is 100MB.
	
	`# snapdrive storage resize -dg testdg -growby 100m -addlun`
	`discovering filer LUNs in disk group testdg...done`
	`LUN toaster:/vol/vol0/testdg_1_SdLun ... created`
	`mapping new lun(s) ... done`
	`discovering new lun(s) ... done.`
	`mapping LUN(s)...done.`
	`initializing LUN(s) and adding to disk group testdg...done`
	`Disk group testdg has been resized`
	`#`

Resizing host volumes and file systems

The `snapdrive storage resize` command applies only to storage system disk groups and volume groups. If you want to increase the size of your host volume or file system, you must use LVM commands. The following table summarizes the LVM commands you can use on the different platforms. For more information on these commands, see their man pages.

Host	Host volume	File systems
AIX	`extendlv`	`chfs`
HP-UX	`lvextend`	`extendfs/fsadm`
Linux	`lvextend`	`resize2fs`
Solaris	`vxassist`	`fsadm`

Connecting LUNs and storage entities to the host

About the storage connect command

The `storage connect` command connects storage entities to the host. Use the `storage connect` command to connect:

- LUNs
- A file system created directly on a LUN
- Disk groups, host volumes and file systems created on LUNs

When you enter the `snapdrive storage connect` command to connect LUNs to the host, SnapDrive for UNIX performs the necessary discovery and mapping. It does not modify LUN contents.

Guidelines for the storage connect command

Follow these guidelines when you use the `snapdrive storage connect` command:

- Storage that includes LVM entities has special requirements. To use the `snapdrive storage connect` command to connect LVM entities, you must create the storage so that each entity in the storage hierarchy has exactly one instance of the next entity. For example, you can use the `storage connect` command to connect a storage hierarchy that has one disk group (dg1) with one host volume (hostvol1) and one file system (fs1). However, you can not use the `storage connect` command to connect a hierarchy that has one disk group (dg1) with two host volumes (hostvol1 and hostvol2) and two file systems (fs1 and fs2).
- On Linux hosts, the `snapdrive storage connect` command connects a file system created directly on a LUN only when the underlying LUN is partitioned.

Connecting LUNs

To use the `storage connect` command to map LUNs to the host, complete the following step.

Step	Action

1	Execute the `snapdrive storage connect` command. This command has the following format:
	`snapdrive storage connect -lun long_lun_name [lun_name ...]`
	`  [-igroup ig_name [ig_name ...]]`
	Note
	For details about using the options and arguments in these command line, see the "SnapDrive for UNIX keywords and arguments" on page 292.
	◆ The first value you supply with the `-lun` option must include the storage system name, volume, and LUN name. To connect multiple LUNs on the same volume, you can use relative pathnames for the `-lun` option after you supply the complete information in the first pathname. When SnapDrive for UNIX encounters a relative pathname, it looks for the LUN on the same volume as the previous LUN. To connect additional LUNs that are not on the same volume, enter the full pathname to each LUN.
	◆ NetApp strongly recommends that you use the default igroup that SnapDrive for UNIX creates to contain the LUNs being connected instead of specifying an igroup on the target storage system.

	Example 1: The following command line connects a single LUN (toaster_lun2) on the storage system filer1 to the host
	`# snapdrive storage connect -lun filer1:/vol/vol1/toaster_lun2`
	Example 2: This command line connects multiple LUNs from the same storage system to the host. Because toaster_lun3 and toaster_lun4 don't specify full pathnames, SnapDrive for UNIX looks for them on the same volume of storage system filer1 as toaster_lun2.
	`# snapdrive storage connect -lun filer1:/vol/vol1/toaster_lun2 toaster_lun3 toaster_lun4`
	Example 3: This command line connects LUNs from different storage systems to the host.
	`# snapdrive storage connect -lun filerA:/vol/vol1/lunA filerB:/vol/vol1/lunB`

Connecting a file system created directly on a LUN

To use the `storage connect` command to connect a file system created directly on a LUN, complete the following step.

Step	Action
1	Execute the `snapdrive storage connect` command. This command has the following format: `snapdrive storage connect -fs file_spec -nolvm -lun long_lun_name [-igroup ig_name [ig_name ...]] [-nopersist]` **Note** For details about using the options and arguments in these command line, see "Examples of storage connect command lines" on page 276 ◆ The *file_spec* given to `-fs` is the name of the file system mount point. ◆ The `-nolvm` argument connects the file system to the LUN without activating the host LVM. This argument is required. ◆ The `-lun` is the name of the LUN that will comprise the file system. The value you supply must include the name of the storage system, volume, and LUN. ◆ The `-nopersist` option allows you to connect a file system without creating an entry in the host file system table (for example, `fstab` on Linux). By default the `storage connect` command creates persistent mounts. This means that when you create an LVM storage entity on a Solaris, AIX, HP-UX or Linux host, SnapDrive automatically creates the storage, mounts the file system and then places an entry for the file system in the host file file system table. ◆ NetApp strongly recommends that you use the default igroup that SnapDrive for UNIX creates to contain the LUNs being connected instead of specifying an igroup on the target storage system.
	Example 1: The following command connects the file system /acct/acctfs on the storage system toaster to a new host. `# snapdrive storage connect -fs /acct/acctfs -lun toaster:/vol/vol1/acctlun1 -nolvm`

Connecting LUNs with disk groups, host volumes and file systems

To use the `storage connect` command to connect LUNs that have disk groups, host volumes and file systems, complete the following step.

Step	Action
1	Execute the `snapdrive storage connect` command. This command has the following format: `snapdrive storage connect -fs file_spec -hostvol file_spec -lun long_lun_name [lun_name ...] [-igroup ig_name [ig_name ...]] [-nopersist]` **Note** For details about using the options and arguments in these command line, see "Checklist for connecting storage" on page 275 ◆ The `-fs file_spec and -hostvol file_spec` you supply identify the LVM file system, disk group and host volumes that you want to connect to a new host. The storage hierarchy that you connect must contain a disk group, host volume and file system, and must conform to the requirements in "Guidelines for the storage connect command" on page 198. You must specify a value for the `-fs` and `-hostvol`. The `-hostvol` must include the name of the disk group. ◆ The `-lun long_lun_name` argument specifies the LUN that will comprise the storage entity. You must include the name of the storage system, volume, and LUN. ◆ The `-nopersist` option allows you to connect a file system without creating an entry in the host file system table (for example, `fstab` on Linux). By default the `storage connect` command creates persistent mounts. This means that when you create an LVM storage entity on a Solaris, AIX, HP-UX or Linux host, SnapDrive automatically creates the storage, mounts the file system and then places an entry for the file system in the host file file system table. ◆ NetApp strongly recommends that you use the default igroup that SnapDrive for UNIX creates to contain the LUNs being connected instead of specifying an igroup on the target storage system.
	Example 1: The following command connects the file system /acct/acctfs in diskgroup `dg1` and host volume `saleshostvol`. All storage entities are contained on the `acctlun1` on the storage system `toaster`. `# snapdrive storage connect -fs /acct/acctfs -hostvol dg1/saleshostvol/ -lun toaster:/vol/vol1/acctlun1`

Disconnecting LUN mappings from the host

Using storage disconnect

The `storage disconnect` operation removes the LUNs, or the LUNs and storage entities that were mapped to the host using the `storage create` or `storage connect` command. Use the `storage disconnect` command to disconnect:

- LUNs
- A file system created directly on a LUN
- Disk groups, host volumes and file systems created on LUNs

When SnapDrive for UNIX removes the LUN mappings, it exports the disk groups or file systems that the LUNs contain. This action, which marks the disk and file system as exported, is the only change that disconnecting the mappings has on the contents of the LUNs.

Methods for disconnecting storage

To make it easier to disconnect the storage, SnapDrive for UNIX provides several formats for the `snapdrive storage disconnect` command. This is because the disconnect operations fall into the following general categories:

- Specifying the LUNs that you want to disconnect from the host. See "Disconnecting LUNs from the host" on page 204 for the steps required to do this.
- Specifying a file system that is created directly on a LUN that you want to disconnect from the host. SnapDrive disconnects both the file system and LUN. See "Disconnecting a file system created on a LUN from the host" on page 205 for the steps required to do this.
- Specifying a disk group, host volume or file system that resides on LUNs you want to disconnect from the host. SnapDrive for UNIX disconnects all the LUNs associated with that entity, and also removes mappings for the file system, host volume, and disk group that comprise the entity you disconnected. See "Disconnecting LUNs and storage entities from the host" on page 206 for the steps required to do this.

Guidelines for the storage disconnect command

Follow these guidelines when using the `snapdrive storage disconnect` command:

- When you disconnect a filesystem, SnapDrive for UNIX always removes the mount point.

Linux hosts allow you to attach multiple fileystems to a single mount point. However, SnapDrive for UNIX requires a unique mount point for each file system. The SnapDrive for UNIX `storage disconnect` command willl fail if you use it to disconnect filesystems that are attached to a single mount point.

- If you use the `-lun` option to specify the name of a LUN that is a member of either a host disk group or a file system, the `host disconnect` command fails.
- If you use `-lun` option to specify the name of the LUN that is not discovered by multipathing software on the host, the s`torage disconnect` command fails. For example, on Solaris hosts, the LUN has to be under DMP control. In other words, the LUN has to have a corresponding /dev/vx/dmp device.

Tips for using storage disconnect

When you use the `storage disconnect` command on some operating systems, you lose information such as the host volume names, the file system mount point, the storage system volume names, and the names of the LUNs. Without this information, reconnecting the storage at a later point in time is difficult.

To avoid this potential problems, NetApp recommends that you first create a snapshot of the storage using the `snapdrive snap create` command **before** you execute the `snapdrive storage disconnect` command.

That way, if you want to reconnect the storage later, you can use the following workaround:

1. Execute the command the following command:

   `snapdrive snap restore` *`filespec`* `-snapname` *`long_snap_name`*

   Include the full path to the snapshot in this command

2. Now you can remove the snapshot, if you choose, by executing the `snapdrive snap delete` command.

Disconnecting LUNs from the host

To use the `storage disconnect` command to remove the mappings for the LUNs you specify, complete the following step.

Step	Action
1	Execute the `snapdrive storage disconnect` command. `snapdrive storage disconnect -lun long_lun_name [lun_name...]` **Note** For details about using the options and arguments in these command line, see the "SnapDrive for UNIX keywords and arguments" on page 292. Keep the following in mind as you use this command: ◆ The first value you supply with the `-lun` option must include the storage system name, volume, and LUN name. To disconnect multiple LUNs on the same volume, you can use relative pathnames for the `-lun` option after you supply the complete information in the first pathname. When SnapDrive for UNIX encounters a relative pathname, it looks for the LUN on the same volume as the previous LUN. To disconnect additional LUNs that are not on the same volume, enter the full pathname to each LUN.

	Example 1: The following command line disconnects a single LUN (toaster_lun2) on the storage system filer1.
	`# snapdrive storage disconnect -lun filer1:/vol/vol1/toaster_lun2`
	Example 2: This command line disconnects LUNs from different storage systems to the host.
	`# snapdrive storage disconnect -lun filerA:/vol/vol1/lunA filerB:/vol/vol1/lunB`

Disconnecting a file system created on a LUN from the host

To use the storage disconnect command to remove a file system created directly on a LUN, complete the following step

Step	Action
1	Execute the `snapdrive storage disconnect` command. `snapdrive storage disconnect -fs file_spec` **Note** For details about using the options and arguments in these command line, see the "SnapDrive for UNIX keywords and arguments" on page 292. Keep the following in mind as you use this command: ◆ The *file_spec* given to `-fs` is the name of the file system mount point. SnapDrive for UNIX automatically locates and disconnects the LUN that is associated with the file system you specify.
	Example 1: The following command line disconnects the file system /mnt/`acc1`. `# snapdrive storage disconnect -fs /mnt/acc1`

Disconnecting LUNs and storage entities from the host

To use the storage disconnect command to remove the mappings for the LUNs associated with storage entities, complete the following step.

Step	Action
1	Execute the `snapdrive storage disconnect` command: `snapdrive storage disconnect { -dg | -fs | -hostvol } file_spec [file_spec ...] [{ -dg | -fs | -hostvol } file_spec [file_spec ...] ...] [-full]` **Note** The `-dg` and `-vg` options are synonyms that reflect the fact that some operating systems refer to disk groups and others refer to volume groups. In addition, `-lvol` and `-hostvol` are also synonyms. This guide uses `-dg` to refer to both disk groups and volume groups and `-hostvol` to refer to both logical volumes and host volumes. For details about using all the options and arguments available with this command, see "SnapDrive for UNIX keywords and arguments" on page 292. Keep the following in mind as you use this command: ◆ This form of the command disables the file system, volume group or disk group you specify and disconnects the LUNs associated with it. ◆ If you include the `-full` option, SnapDrive for UNIX disconnects host-side entities that contain other entities, such as a disk group containing one or more host volumes. Without the `-full` option, this command fails if any of the specified entities contain other entities. For example, the command fails if there are disk groups that contain host volumes or host volumes that contain file systems.
	Example 1: The following command line disconnects the disk group dg1. `# snapdrive storage disconnect -dg dg1`

Connecting only the host-side of storage

How the host connect command works

The `snapdrive host connect` command performs the host side operations that connect a storage entity. Use the `snapdrive host connect` command to connect

- LUNs
- A file system created directly on a LUN
- Disk groups, host volumes and file systems created on LUNs

When you enter the `snapdrive storage connect` command to connect LUNs to the host, SnapDrive for UNIX performs the necessary discovery and mapping. It does not modify LUN contents.

Connecting LUNs: When you use the `snapdrive host connect` command to connect LUNs, SnapDrive assumes that the LUNs have been fully provisioned and mapped to the host. The command performs only the host operations (discovery) required to connect the LUN.

Connecting LUNs that contain storage entities: When you use the `snapdrive host connect` command to connect a LUN that has a file system, host volume, or disk group, SnapDrive assumes that the LUN has been created on a storage system and mapped to an igroup. The command performs the host operations (discovery, file system mounting, etc) required to connect the storage.

Note

The storage entity you connect must conform to specific configuration requirements. Refer to "Guidelines for the host connect command" below, for additional information

Guidelines for the host connect command

Follow these guidelines when connecting a snapshot to the host:

- Storage that has LVM entities has special requirements. To use the `snapdrive host connect` command to connect LVM entities, you must create the storage so that each entity in the storage hierarchy has exactly one instance of the next entity. For example, you can use the `snapdrive host connect` command to connect a storage hierarchy that has one disk group (dg1) with one host volume (hostvol1) and one file system (fs1). However, you can not use the `snapdrive host connect` command to connect a hierarchy that has one disk group (dg1) with two host volumes (hostvol1 and hostvol2) and two file systems (fs1 and fs2).

- On Linux hosts, the `snapdrive host connect` command connects a file system created directly on a LUN only when the underlying LUN is partitioned.

Connecting LUNs

To use the `host connect` command to map LUNs to the host, complete the following step.

Step	Action

1	Execute the `snapdrive host connect` command:
	`snapdrive host connect -lun long_lun_name [lun_name ...]`
	Note
	For details about using the options and arguments in these command line, see the "SnapDrive for UNIX keywords and arguments" on page 292.
	Keep the following in mind as you use this command:
	◆ Specify the LUNs that you want to connect to the host. The first value you supply with the `-lun` option must include the storage system name, volume, and LUN name. To connect multiple LUNs on the same volume, you can use relative pathnames for the `-lun` option after you supply the complete information in the first pathname. When SnapDrive for UNIX encounters a relative pathname, it looks for the LUN on the same volume as the previous LUN. To connect additional LUNs that are not on the same volume, enter the full pathname to each LUN.
	Example: The following command line connects the LUNs lunA and lunB.
	`# snapdrive host connect -lun acctfiler:/vol/vol1/lunA lunB`

Connecting a file system created directly on a LUN

To use the `host connect` command to connect a LUN with a file system created directly on it, complete the following step.

Step	Action
1	Execute the `snapdrive host connect` command. This command has the following format: `snapdrive host connect -fs file_spec -nolvm -lun long_lun_name [-nopersist]` **Note** For details about using the options and arguments in these command line, see "Checklist for using the host connect command" on page 279 ◆ The `-fs` *file_spec* for the file system that you want to connect to a new host. ◆ The `-nolvm` arguement connects the file system to the host without activating the host LVM. This arguement is required. ◆ The `-lun long_lun_name` argument specifies the LUN that will comprise the storage entity. You must include the name of the storage system, volume, and LUN. ◆ The `-nopersist` option allows you to connect a file system without creating an entry in the host file system table (for example, `fstab` on Linux). By default the `host connect` command creates persistent mounts. This means that when you create an LVM storage entity on a Solaris, AIX, HP-UX or Linux host, SnapDrive automatically creates the storage, mounts the file system and then places an entry for the file system in the host file file system table.
	Example 1: The following command connects the file system `my_fs that resides on acctlun1` on the storage system `toaster`. `# snapdrive host connect -fs /my_fs -lun toaster:/vol/vol1/acctlun1 -nolvm`

Connecting LUNs with disk groups, host volumes and file systems

To use the `host connect` command to connect LUNs that have disk groups, host volumes and file systems, complete the following step.

Step	Action
1	Execute the `snapdrive host connect` command. This command has the following format: `snapdrive host connect -fs` *`file_spec`* `-hostvol` *`file_spec`* `-lun` *`long_lun_name`* `[`*`lun_name`*`][-nopersist]` **Note** For details about using the options and arguments in these command line, see "Checklist for connecting storage" on page 275 ◆ The `-fs` *`file_spec`* and `-hostvol` *`file_spec`* you supply identifies the LVM file system, disk group and host volumes that you want to connect to a new host. The storage hierarchy that you connect must contain a disk group, host volume and file system, and must conform to the requirements in "Guidelines for the storage connect command" on page 198. You must specify a value for the `-fs` and `-hostvol`. The `-hostvol` must include the name of the disk group. ◆ The `-lun long_lun_name` argument specifies the LUN that will comprise the storage entity. You must include the name of the storage system, volume, and LUN. ◆ The `-nopersist` option allows you to connect a file system without creating an entry in the host file system table (for example, `fstab` on Linux). By default the `host connect` command creates persistent mounts. This means that when you create an LVM storage entity on a Solaris, AIX, HP-UX or Linux host, SnapDrive automatically creates the storage, mounts the file system and then places an entry for the file system in the host file file system table.
	Example 1: The following command connects the file system /acct/ac`ctfs` in diskgroup `dg1` and host volume `saleshostvol`. All storage entities are contained on the `acctlun1` on the storage system `toaster`. `# snapdrive host connect -fs /acct/acctfs -hostvol dg1/saleshostvol/ -lun toaster:/vol/vol1/acctlun1`

Disconnecting only the host-side of storage

Disconnecting host entities from storage

The `snapdrive host disconnect` command disconnects the host-side entities you specify without changing the state on the storage system where the entities reside. Use the snapdrive host disconnect command to disconnect:

- LUNs
- A file system created directly on a LUN
- Disk groups, host volumes and file systems created using the LVM

This command performs the host portion of the `storage disconnect` actions. Because it does not unmap or delete the LUNs, you do not risk disconnecting other hosts from the storage entity if all the hosts share an igroup on the storage system. It only affects the host.

Methods for disconnecting host-side storage

To make it easier to disconnect storage from the host, SnapDrive for UNIX provides several formats for the snapdrive storage disconnect command. This is because storage disconnect operations fall into the following general categories:

- Specifying the LUNS you want to disconnect from the host. See "Disconnecting LUNs from the host" on page 212 for the steps required to do this.
- Specifying the file system created directly on the host that you want to disconnect. See "Disconnecting a file system created on a LUN from the host" on page 213 for the steps required to do this.
- Specifying the LVM disk groups, file systems or host volumes that you want to disconnect. See "Disconnecting LUNs and storage entities from the host" on page 215 for the steps required to do this.

Guidelines for the host disconnect command

Follow these guidelines when using the snapdrive host disconnect command:

- When you disconnect a filesystem, SnapDrive for UNIX always removes the mount point.

  Linux hosts allow you to attach multiple fileystems to a single mount point. However, SnapDrive for UNIX requires a unique mount point for each file system. The SnapDrive for UNIX `host disconnect` command willl fail if you use it to disconnect filesystems that are attached to a single mount point.

- If you use the `-lun` option to specify the name of a LUN that is a member of either a host disk group or a file system, the `host disconnect` command fails.
- If you use `-lun` option to specify the name of the LUN that is not discovered by multipathing software on the host, the s`torage delete` command fails. For example, on Solaris hosts, the LUN has to be under DMP control. In other words, the LUN has to have a corresponding /dev/vx/dmp device.

Tips for using host disconnect

When you use the `snapdrive host disconnect` command on some operating systems, you lose information such as the host volume names, the file system mount point, the storage system volume names, and the names of the LUNs. Without this information, reconnecting the storage at a later point in time is difficult.

To avoid this potential problem, NetApp strongly recommends that you first create a snapshot of the storage using the `snapdrive snap create` command **before** you execute the `snapdrive host disconnect` command.

That way, if you want to reconnect the storage later, you can use the following workaround:

1. Execute the following command:

   `snapdrive snap restore` *`filespec`* `-snapname` *`long_snap_name`*

   Include the full path to the snapshot in this command

2. Now you can remove the snapshot, if you choose, by executing the `snapdrive snap delete` command.

Disconnecting LUNs from the host

To use the `snapdrive host disconnect` command to disconnect LUNs from the host, complete the following step.

Step	Action
1	Execute the `snapdrive host disconnect` command. `snapdrive host disconnect -lun long_lun_name [lun_name...]` **Note** For details about using the options and arguments in these command line, see the "SnapDrive for UNIX keywords and arguments" on page 292. Keep the following in mind as you use this command: ◆ The first value you supply with the `-lun` option must include the storage system name, volume, and LUN name. To disconnect multiple LUNs on the same volume, you can use relative pathnames for the `-lun` option after you supply the complete information in the first pathname. When SnapDrive for UNIX encounters a relative pathname, it looks for the LUN on the same volume as the previous LUN. To disconnect additional LUNs that are not on the same volume, enter the full pathname to each LUN.

	Example 1: The following command line disconnects a single LUN (toaster_lun2) on the storage system filer1.
	`# snapdrive host disconnect -lun filer1:/vol/vol1/toaster_lun2`
	Example 2: This command line disconnects LUNs from different storage systems to the host.
	`# snapdrive host disconnect -lun filerA:/vol/vol1/lunA filerB:/vol/vol1/lunB`

Disconnecting a file system created on a LUN from the host

To use the `snapdrive host disconnect` command to remove a file system created directly on a LUN, complete the following step.

Step	Action
1	Execute the `snapdrive host disconnect` command. `snapdrive host disconnect -fs file_spec` **Note** For details about using the options and arguments in these command line, see the "SnapDrive for UNIX keywords and arguments" on page 292. Keep the following in mind as you use this command: ◆ The *file_spec* given to `-fs` is the name of the file system mount point. SnapDrive for UNIX automatically locates and disconnects any host volume, disk group, or LUN that is associated with the file system you specify.
	Example 1: The following command line disconnects a file system `acc1`. `# snapdrive host disconnect -fs /mnt/acc1`

Disconnecting LUNs and storage entities from the host

To use the `snapdrive host disconnect` command to disconnect a LUN with an LVM entity, complete the following step.

Step	Action

1	Execute the `snapdrive host disconnect` command:
	`snapdrive host disconnect {-dg | -fs | -hostvol} file_spec [file_spec ...]`
	`[{ -dg | -fs | -hostvol } file_spec [file_spec ...] ...] [-full]`
	Note
	For details about using the options and arguments in these command line, see the "SnapDrive for UNIX keywords and arguments" on page 292.
	Keep the following in mind as you use this command:
	◆ You can disconnect an LVM disk group, file system, or host volume. Enter the name of the entity as the value for the *file_spec* argument. The command disables the entity you specify and disconnects the LUNs associated with it.
	Note
	The `-dg` and `-vg` options are synonyms that reflect the fact that some operating systems refer to disk groups and others refer to volume groups. In addition, `-lvol` and `-hostvol` are also synonyms. This guide uses `-dg` to refer to both disk groups and volume groups and `-hostvol` to refer to both logical volumes and host volumes. For details about using all the options and arguments available with this command, see "SnapDrive for UNIX keywords and arguments" on page 292.
	◆ If you include the `-full` option, SnapDrive for UNIX disconnects host-side entities that contain other entities, such as a disk group containing one or more host volumes. Without the `-full` option, this command fails if any of the specified entities contain other entities. For example, the command fails if there are disk groups that contain host volumes or host volumes that contain file systems.
	Example 1: The following command line disconnects the disk groups dg1, dg2, and dg3.
	`# snapdrive host disconnect -dg dg1 dg2 dg3`
	Example 2: This command line disconnects the logical volume dg1/mylvoll1.
	`# snapdrive host disconnect -lvol dg1/mylvol1`

Deleting storage from the host and storage system

Using storage delete

The `snapdrive storage delete` command removes the storage entities on the host as well as all underlying host-side entities and storage system LUNs backing them.

Note

This command deletes data. Use caution in running it.

Methods for deleting storage

To make it easier to delete the storage, SnapDrive for UNIX provides several formats for the `snapdrive storage delete` command. This is because the delete operations fall into the following general categories:

- Specifying the LUNs that you want to delete from the host. See "Deleting specific LUNs" on page 217 for the steps required to do this.
- Specifying the host entity for which you want to delete the storage, including all associated objects and all storage system LUNs. See "Deleting specific host entities" on page 218 for the steps required to do this.

Guidelines for using the storage delete command

The `snapdrive storage delete` command has the following restrictions in SnapDrive for UNIX:

- When you delete a filesystem, SnapDrive for UNIX always removes the file system's mount point.

  Linux hosts allow you to attach multiple fileystems to a single mount point. However, SnapDrive for UNIX requires a unique mount point for each file system. The SnapDrive for UNIX `storage delete` command willl fail if you use it to delete filesystems that are attached to a single mount point.
- If you use the `-lun` option to specify the name of a LUN that is a member of either a host disk group or a file system, the `storage delete` command fails.
- If you use `-lun` option to specify the name of the LUN that is not discovered by multipathing software on the host, the `storage delete` command fails. For example, on Solaris hosts, the LUN has to be under DMP control. In other words, the LUN has to have a corresponding /dev/vx/dmp device.

Deleting specific LUNs

To delete specific LUNs, complete the following step.

Step	Action

1	Execute the `snapdrive storage delete` command.
	`snapdrive storage delete -lun long_lun_name [lun_name...]`
	Note
	For details about using the options and arguments in these command line, see the "SnapDrive for UNIX keywords and arguments" on page 292.
	Keep the following in mind as you use this command.
	◆ This form of the command deletes the LUNs you specify. You can use the command to delete LUNs on a specific storage system and volume or on multiple storage systems.
	Note
	This command deletes data. Use caution in running it.
	◆ The first value you supply with the -lun option must include the storage system name, volume, and LUN name. To delete multiple LUNs on the same volume, you can use relative pathnames for the -lun option after you supply the complete information in the previous pathname. When SnapDrive for UNIX encounters a relative pathname, it looks for the LUN on the same volume as the previous LUN. To delete additional LUNs that are not on the same volume, enter the full pathname to each LUN.
	Example 1: The following command line tells SnapDrive for UNIX to delete LUNs from storage systems.
	`# snapdrive storage delete -lun filer1:/vol/vol1/lun1 lun2`

Deleting specific host entities

To delete a storage entity such as a disk group and its associated LUNs, or a file system that resides directly on a LUN, complete the following step.

Step	Action

1	Execute the `snapdrive storage delete` command.
	`snapdrive storage delete {-dg | -fs | -hostvol } file_spec [file_spec ...] [{-dg | -fs | -hostvol } file_spec [file_spec ...]] [-full]`
	Note
	For details about using the options and arguments in these command line, see the "SnapDrive for UNIX keywords and arguments" on page 292.
	Keep the following in mind as you use this command:
	◆ This form of the command deletes all the host and storage system components for the storage entities that you specify.
	◆ If you include the `-full` option, SnapDrive for UNIX deletes host-side entities that contain other entities, such as a disk group containing one or more host volumes. Without the `-full` option, this command fails if any of the specified entities contain other entities. For example, the command fails if there are disk groups that contain host volumes or host volumes that contain file systems.
	Example: The following command lines tell SnapDrive for UNIX to delete the objects from specific host entities, such as disk groups, logical volumes and file systems.
	`# snapdrive storage delete -full -dg dg1 dg2 dg3`
	`# snapdrive storage delete -lvol mydg/vol3 mydg/vol5`
	`# snapdrive storage delete -fs /mnt/acct2`

Troubleshooting 7

About this chapter

This chapter provides information about the troubleshooting tools available. These tools are useful for gathering information as well as solving problems.

Topics in this chapter

This chapter discusses the following topics:

- "Data collection utility" on page 220
- "General troubleshooting tips for SnapDrive for UNIX" on page 223
- "Understanding error messages" on page 227
- "Common error conditions" on page 229
- "Standard exit status values" on page 231

Data collection utility

About the data collection utility

SnapDrive for UNIX provides a data collection utility (snapdrive.dc) that collects diagnostic information about SnapDrive for UNIX and your system setup. It does this by running diagnostic utilities and copying SnapDrive for UNIX log files to a special directory. Then it creates a compressed file containing this information that you can send to technical support for analysis.

Note

This utility only gathers basic information about the system and the configuration of SnapDrive for UNIX. It does not copy the file containing login information for the storage systems. It also does not make any configuration changes.

Tasks performed by snapdrive.dc

The snapdrive.dc utility performs the following tasks:

- Runs the <*host*>_info utility, which comes with your host attach or support kit. For example, the Solaris kit includes the solaris_info utility, while the HP-UX kit includes the hpux_info utility. On AIX, it is the aix_info utility and on Linux, it is the linux_info utility. The <*host*>_info utility provides information about the host and saves the information to a compressed file.
- Creates a directory called <*directory*>/ontap_snapdrive_<*name*> (the directory pathname can vary depending your host; see the kit documentation for more information on this pathname). It places copies of the following files in it:
  - SnapDrive version, as indicated by running the `snapdrive version` command
  - The snapdrive.conf file
  - The audit log file
  - The trace log file
  - The recovery log file
  - The snapdrive.conf file
  - The file created by the <*host*>_info utility
- Creates a compressed file of the directory contents and displays a message suggesting you send this file to technical support.

Executing the data collection utility

To execute the data collection utility, complete the following steps.

Step	Action
1	Log in as root.
2	Change to the SnapDrive for UNIX diagnostic directory. The path is: `<install_directory>/diag` *install_directory* is the SnapDrive for UNIX installation directory for your host operating system. This directory can vary depending on your host operating system. See the installation steps to determine where this directory is on your host.
3	At the command prompt, enter the following command: **`snapdrive.dc [-d <directory>] [-n <file_name>] [-f]`** `-d` <*directory*> specifies the location for the compressed file that this utility creates. The default location is /tmp/ontap. `-n` <*file_name*> specifies a string to be included in the name for the directory and compressed output file. If you supply a value for this argument, the snapdrive.dc utility creates a directory called ontap_snapdrive_<*name*> and a file name called ontap_snapdrive_<*name*>.tar.Z. The default path name is /tmp/ontap/ontap_snapdrive_info.tar.Z. `-f` forces SnapDrive for UNIX to overwrite the files if they currently exist. Without the `-f` option, the snapdrive.dc utility returns an error message if the output directory or the file name exists. It does not overwrite any directories or file names. **Result:** This utility runs the <*host*>_info script for your host operating system, as well as copying log files and the configuration file for SnapDrive for UNIX. It places this information in a directory, and then creates a compressed file containing the contents of the directory.
4	Send the <*directory*>/ontap_snapdrive_<*name*>.tar.Z file to technical support for analysis.

Examples of using snapdrive.dc

The following are examples of using the snapdrive.dc utility.

Example 1: This example runs the snapdrive.dc utility without supplying any command-line arguments:

```
# snapdrive.dc
SnapDrive configuration info and logs are in directory
/tmp/ontap/ontap_snapdrive_info.
Compressed file is /tmp/ontap/ontap_snapdrive_info.tar.Z.
Please send this file to technical support for analysis.
```

Example 2: This example uses the command-line options to specify a directory and a name for the resulting file:

```
# snapdrive.dc -d . -n mysystem
SnapDrive configuration info and logs are in directory
./ontap_snapdrive_mysystem.
Compressed file is ./ontap_snapdrive_mysystem.tar.Z.
Please send this file to technical support for analysis.
```

General troubleshooting tips for SnapDrive for UNIX

Attempts to create a snapshot fail

The following conditions can affect the `snapdrive snap create` command's ability to create a snapshot:

- The `snapdrive snap create` command must be able to perform the following tasks:
  - Query all disk groups for a list of LUNs
  - Query all LUNs via SCSI commands
- The LUNs in the disk group must be online and mapped to the disk group before you take the snapshot. If any of the LUNs are offline or unmapped, the create operation fails.

  Make sure that all the LUNs are online and mapped to the host before you attempt to take a snapshot.
- The access permissions do not allow the host to create a snapshot of information on that storage system. For more information on access permissions, see "Setting up access control" on page 105.
- Make sure you are logged in as root. SnapDrive for UNIX requires this for its commands. In addition, because it works with LVM entries, you must be logged on to the host with sufficient permission for it to manipulate the LVM entries.

Attempts to restore a snapshot fail

The following can affect the `snapdrive snap restore` command's ability to restore a snapshot:

- The `snapdrive snap restore` command must be able to perform the following tasks:
  - Query all disk groups for a list of LUNs
  - Query all LUNs via SCSI commands

  The host volumes and file systems do not need to be available and mounted for the `snapdrive snap restore` command to succeed.
- The access permissions must allow the host to create and restore a snapshot of information on that storage system. For more information on access permissions, see "Setting up access control" on page 105.
- The NFS mounted directory must be exported correctly to the host, so that it can be mounted. Refer to "NFS considerations" on page 18 for additional information.

Caution

You should always take another snapshot after you add a LUN to or remove a LUN from a disk group.

- If you add a host volume or file systems to a disk group after taking a snapshot and then try to restore the snapshot, you receive an error message. You must include the force option (`-f`) on the command line for the restore operation to succeed in this situation.

  Caution

  Host volumes and file systems that you add after taking a snapshot are no longer accessible when you restore the snapshot.

- If you rename a host volume or a file system or change the mount point after taking a snapshot and then try to restore it, the restore operation fails. You must include the force option (`-f`) on the command line for the restore operation to succeed in this situation.

  Caution

  Use the `-f` option with great care to make sure you don't accidentally overwrite something that you didn't intend to overwrite.

- If you rename a LUN after taking a snapshot and then try to restore that snapshot, the restore operation fails. After you rename a LUN, you should take a new snapshot.

WARNING

After you start a `snapdrive snap restore` operation, do not halt it. Halting this operation might leave the system in an inconsistent state. You then might need to perform a manual recovery.

How SnapDrive commands mounts file systems, and modify system files

Certain SnapDrive for UNIX commands can cause file systems to be mounted or unmounted. When a command performs an operation that mounts a file system, SnapDrive for UNIX adds the file system's name to the standard system file for your host platform. If an operation unmounts a file system, SnapDrive for UNIX removes the name from the system file. This should not pose a problem; however, it is useful to know when system files are modified.

The name of the system file varies depending on your host platform. The following table lists the host platforms and their system files.

Host	System file
AIX	etc/filesystems
HP-UX or Linux	/etc/fstab
Solaris	/etc/vfstab

Any of the following commands can mount file systems:

- `snap restore`
- `snap connect`
- `storage create`

The following commands perform operations that can unmount a file system:

- `snap restore`
- `snap disconnect`
- `storage disconnect`
- `storage delete`
- `host disconnect`

Some commands delayed when storage cluster is in failover mode

SnapDrive for UNIX commands that create, connect, delete, or disconnect a LUN take several minutes to complete on Solaris host while one of the clustered storage systems is down.

Such commands are `storage create`, `storage connect`, `storage delete`, `storage disconnect`, `snap connect`, `snap disconnect` and, in some cases, `snap restore`.

This problem affects `snap restore` only when SnapDrive for UNIX needs to restore LUNs that were deleted or disconnected.

Data distribution on disks changes after a restore operation

Symptom: The data distribution on a disk changes after you restore a snapshot.

Explanation: The restore operation replaces the data on a disk based on the way it was when you took the snapshot. If you take a snapshot and then change the way data is distributed among disks in the disk group, the restored snapshot returns the data to the previous disk configuration. This means that changes to striping, mirroring, and so on can take place as a result of a restore operation.

What to do: If you change the data distribution on the disks in the disk group, take a new snapshot.

What to do: You can use the parameter `rbac-vol-name` in the snapdrive.conf file to specify a different volume name where SnapDrive for UNIX looks for the permissions file. Specify the name of the volume with no /vol suffix, as in:

```
rbac-vol-name="vol1"
```

SnapDrive for UNIX then uses that volume name for all the storage systems accessed from the local host.

Understanding error messages

Error message locations

SnapDrive for UNIX provides information about error messages in the following places:

- The command line. It displays all messages to the standard error output of the SnapDrive for UNIX command.
- The system log. SnapDrive for UNIX logs all errors that have a severity level of Fatal and Admin error to the system log using the syslog(3) mechanism.

In addition, SnapDrive for UNIX records information about the commands and their execution in the log files, specifically:

- The audit log file. The audit log records the following information for each SnapDrive for UNIX command:
  - Who issued it.
  - When it was issued.
  - What its exit status was. This is very useful in determining what actually happened on a system.
- The trace log file. The trace log records more detailed information about any errors that occur. Technical support uses this log when diagnosing problems.

Error message format

SnapDrive for UNIX error messages conform to the following format:

```
<message-ID> <error type>: message text
```

where:

message-ID - a unique identifier used by technical support to locate the specific code that produced the error. If you need to call technical support, it is recommended that you record the message-ID that accompanied the error message.

error type - specifies the type of error that SnapDrive encountered. Return values include:

Warning - Snapdrive for UNIX executed the command but issued a warning about conditions that might require your attention.

Command - Snapdrive for UNIX failed to execute the command due to an error in the command line. Check the command line format and variables to ensure they are correct.

Admin - Snapdrive for UNIX failed to execute the command due to incompatibilities in the system configuration. Contact your System Administrator to review your configuration parameters.

Fatal - Snapdrive for UNIX failed to execute the command due to an unexpected condition. Fatal errors are rare. If a fatal error occurs and you have problems resolving it, contact technical support for assistance in determining the steps you need to take to recover correctly and fix any error condition.

message text - information that explains the error. This text might include information from another component to provide more detail about the error. For example, if a command-line argument such as a disk group is missing, the error message tells you what is missing. Or the ManageONTAP APIs that SnapDrive for UNIX uses to control the storage system might supply additional text to help explain the error. In this case, the text follows the basic SnapDrive for UNIX error message.

Sample error message

The following message indicates a problem on the command line. The message-ID is 0001-426.

```
0001-426 Command error: File system mount point /u/rsmith/.bashrc
is not available for use. Please choose another mount point
```

Common error conditions

Introduction

This section provides information about common error messages that you may encounter. This chapter does not include information for every component or error condition.

Operating system limits on open files

SnapDrive for UNIX checks for operating system limitations on the number of files opened by a process.

Note

The default limit for the number of file handles opened simultaneously by one process varies based on your operating system. Check your operating system documentation to determine the limit.

If the number of open LUNs for one operation exceeds the operating system limit on the number of file handles opened simultaneously by one process, SnapDrive exits with the following error message:

```
0001-001 Admin error: Unable to open device <path-to-device>
```

Example: You see an error message similar to the following one if this limit is exceeded on a Solaris host:

```
0001-001 Admin error: Unable to open device /dev/rdsk/c1t1d26s2
```

SnapDrive for UNIX reports an error if iSCSI is not running

Symptom: SnapDrive for UNIX running on a Linux host returns an error message stating that the device is not a correct LUN. This happens after one of the following situations occurs:

- An iSCSI stop was issued.
- A takeover has happened and the iSCSI service hasn't started yet.
- A giveback is happening and the iSCSI service hasn't started yet.

Explanation: For SnapDrive for UNIX to execute, the iSCSI service must be running. If the service is halted, SnapDrive for UNIX returns an error message. For example, if you executed a `snapdrive storage show` command when the iSCSI service is not running, you might see output similar to the following:

```
# snapdrive storage show -dg toaster1

dg: toaster1
hostvol: /dev/toaster1/lvol1     state: AVAIL
hostvol: /dev/taoster1/lvol2     state: AVAIL
fs: /dev/toaster1/lvol1  mount point: /mnt/um1
fs: /dev/toaster1/lvol2  mount point: NOT MOUNTED

 device filename adapter   path  size     proto   state   lun path
-----------------------------------------------------------
/dev/sdb          -        P     34:device  fcp     /dev/sdb is not correct LUN ()

/dev/sdc          -        P     34:device  fcp     /dev/sdc is not correct LUN ()


0001-185 Command error: storage show failed: no correct devices to show
```

What to do: Make sure the iSCSI service is running. If the storage system is in the process of a takeover or giveback, wait until it starts the iSCSI service. If the storagea system is in normal running mode, restart the iSCSI service. Once the service is running, reissue the `snapdrive storage show` command. This time you should see output similar to the following:

```
# snapdrive storage show -dg toaster1

dg: toaster1
hostvol: /dev/toaster1/lvol1     state: AVAIL
hostvol: /dev/toaster1/lvol2     state: AVAIL
fs: /dev/toaster1/vol1  mount point: /mnt/um1
fs: /dev/toaster1/lvol2  mount point: NOT MOUNTED

 device filename        adapter path size    proto    state   lun path
-----------------------------------------------------------
/dev/sdb                  -      P    2g      fcp     online  eccentric:/vol/vol1/lun1
/dev/sdc                  -      P    2g      fcp     online  eccentric:/vol/vol1/lun2
#
```

Standard exit status values

Understanding exit status values of error messages

Each SnapDrive error message ID is linked to an exit status value. Exit status values contain the following information:

Exit status value - indicates the basic cause of the error condition.

Type - indicates the type of error. The level of seriousness depends on the message, not the value. Possible values are:

Warning - Snapdrive for UNIX executed the command but issued a warning about conditions that might require your attention.

Command - Snapdrive for UNIX failed to execute the command due to an error in the command line. Check the command line format to ensure they are correct.

Admin - Snapdrive for UNIX failed to execute the command due to incompatibilities in the system configuration. Contact your System Administrator to review your configuration parameters.

Fatal - Snapdrive for UNIX failed to execute the command due to an unexpected condition. Fatal errors are rare. If a fatal error occurs and you have problems resolving it, contact technical support for assistance in determining the steps you need to take to recover correctly and fix any error condition.

Using exit status values

Exit status values are used in scripts to determine the success or failure of an SnapDrive command.

- A value of zero indicates that the command completed successfully.
- A non-zero value indicates that the command did not complete, and provides information about the cause and severity of the error condition.

Script example

The following script uses SnapDrive for UNIX exit status values:

```
#!/bin/sh
# This script demonstrates a SnapDrive
# script that uses exit codes.

RET=0;
#The above statement initializes RET and sets it to 0
```

```
snapdrive snap create -dg vg22 -snapname vg22_snap1;
# The above statement executes the snapdrive command

RET=$?;
#The above statement captures the return code.
#If the operation worked, print
#success message. If the operation failed, print
#failure message and exit.

if [ $RET -eq 0 ]; then
echo "snapshot created successfully"
else
echo "snapshot creation failed, snapdrive exit code was $RET"
exit 1
fi

exit 0;
```

If RET=0, the command executed successfully and the script outputs the following:

```
# ./tst_script
snap create: snapshot vg22_snap1 contains:
disk group vg22 containing host volumes
lvol1
snap create: created snapshot betty:/vol/vol2:vg22_snap1
snapshot created successfully
```

If RET= a non-zero value, the command did not execute successfully. The following example shows typical output.

```
# ./tst_script
0001-185 Command error: Snapshot betty:/vol/vol2:vg22_snap1 already
exists on betty:/vol/vol2.
Please use -f (force) flag to overwrite existing snapshot
snapshot creation failed, snapdrive exit code was 4
```

Exit status values

The following table contains information about exit status values. The exit status values are numbered sequentially. If SnapDrive for UNIX does not currently implement an error, that exit status value is not included in the table. As result, there may be some gaps in the numbers.

Exit value	Error name	Type	Description
1	Not supported	Command error	A function was invoked that is not supported in this version of SnapDrive for UNIX.
2	No memory	Fatal	The system has run out of memory. SnapDrive for UNIX cannot proceed until you free enough memory for it to work. Check other applications running to verify that they are not consuming excessive memory.
3	Invalid command	Command error	You issued an invalid command; this is likely to be a syntax error in the text of the command you entered.
4	Already exists	Command error	You requested that something be created that already exists. Usually, this error refers to a snapshot name, which must not exist on the storage system volume where you are taking the snapshot.
5	Create thread failed	Admin error	SnapDrive for UNIX could not create a process thread. Check the other processes running on the system to make sure that enough thread resources are available.
6	Not found	Command error	You included a file, data group, host volume, file system, or other argument on the SnapDrive for UNIX command line that does not exist.
7	Not a mounted file system	Command error	The file system you want to access either is not a valid file system or is not mounted.
9	Volume manager error	Command error	An error was returned when accessing the volume manager. See the specific error message to get details of which error, and why.

Exit value	Error name	Type	Description
10	Invalid name	Command error	You supplied a name on the command line that was not correctly formatted. For example, a storage system volume was not specified as "<*filer*>:/vol/<*volname*>". This message also occurs when an invalid character is given in either a storage system or a volume manager-based name.
11	Device not found	Admin error	SnapDrive for UNIX cannot access a LUN in the disk group that you want to take a snapshot of. Check the status of all LUNs, both on the host and on the storage system. Also check that the storage system volume is online, and that the storage system is up and connected to the host.
12	Busy	Command error	The LUN device, file, directory, disk group, host volume, or other entity was busy. This is generally a nonfatal error that goes away when you retry the command. It sometimes indicates that a resource or process is hung, causing the object to be busy and unavailable for SnapDrive for UNIX to use. It could also indicate you are trying to take a snapshot during a period when there is too much heavy I/O traffic for the snapshot to succeed.
13	Unable to initialize	Fatal	SnapDrive for UNIX could not initialize third-party material that it needs. This can refer to file systems, volume managers, cluster software, multipathing software, and so on.

Exit value	Error name	Type	Description
14	SnapDrive busy	Command error	Another user or process is performing an operation on the same hosts or storage systems at the same time you asked SnapDrive for UNIX to perform an operation. Retry your operation. Occasionally this message means that the other process is hung and you must kill it. **Note** The `snap restore` operation can take a long time under some circumstances. Be sure that the process you think is hung is not just waiting for a `snap restore` operation to be completed.
15	Config file error	Fatal	The snapdrive.conf file has invalid, inadequate, or inconsistent entries. See the specific error message for details. You must correct this file before SnapDrive for UNIX can continue. See "Setting values in snapdrive.conf" on page 85 for information about modifying the snapdrive.conf file.
17	Bad permissions	Command error	You do not have permission to execute this command. You must be logged in as root to run SnapDrive for UNIX.
18	No filer	Admin error	SnapDrive for UNIX cannot contact the storage system needed for this command. Check the connectivity to the storage system indicated in the error message.
19	Bad filer login	Admin error	SnapDrive for UNIX cannot log in to the storage system using the login information you supplied.
20	Bad license	Admin error	A service SnapDrive for UNIX requires is not licensed to run on this storage system.

Exit value	Error name	Type	Description
22	Cannot freeze fs	Admin error	A `snap create` operation failed because SnapDrive for UNIX could not freeze the file systems specified in order to make the snapshot. Confirm that the system I/O traffic is light enough to freeze the file system and then retry the command.
26	Resource busy on restore	Admin error	The `snap restore` operation failed because a resource is busy that must be disconnected or disabled before the operation can begin. Examples of potentially busy resources include a file system or logical volume. Either repeat the command and include the `-f` (force) option on the command line or terminate the processes using the resource. **Note** On some systems, the `-f` option does not disconnect the resource.
27	Inconsistent snapshot	Admin error	The `snap restore` operation failed because you requested a restore from a snapshot with inconsistent images of the disk group. Inconsistent images can occur in the following cases: ◆ You did not take the snapshot using SnapDrive for UNIX. ◆ The `snap create` operation was interrupted before it set consistent bits and, thus, could not clean up (as in the case of a catastrophic system failure). ◆ Some type of data problem occurred with the snapshot after it was taken.
28	HBA failure	Admin error	SnapDrive for UNIX encountered an error while trying to retrieve information from the HBA.

Exit value	Error name	Type	Description
29	Bad metadata	Admin error	SnapDrive for UNIX encountered an error in the snapshot metadata that it wrote when it created the snapshot.
30	No snapshot metadata	Admin error	SnapDrive for UNIX can't perform a restore operation because the metadata does not contain all requested disk groups.
31	Bad password file	Admin error	The password file has a bad entry. Use the `snapdrive config delete` command to delete the login entry for this storage system. Then reenter the login information using the `snapdrive config set user_name` command. For information on specifying user logins to storage systems, see "Specifying the current login information for storage systems" on page 111.
32	No password file	Admin error	Either there is no password file containing the storage system login information or the path to the password file is incorrect. Check the path given for the password file in the snapdrive.conf file and make sure that the password file has been initialized with logins for your storage systems. You can use the `snapdrive config show` command to display the configuration file. For information on updating this file, see "Setting values in snapdrive.conf" on page 85.
33	No password file entry	Admin error	The password file has no entry for this storage system. Run snapdrive config set *user_name* <*filer_name*> for every storage system on which you need to run SnapDrive for UNIX. Then try this operation again.
34	Not a LUN	Admin error	A SnapDrive for UNIX command encountered a LUN that is not on a correct storage system.

Exit value	Error name	Type	Description
35	User aborted	Admin error	The system displayed a prompt asking you to confirm an operation and you indicated that you did not want the operation performed.
36	I/O stream error	Admin error	The system input or system output routines returned an error that SnapDrive did not understand. Run snapdrive.dc and send that information to technical support so that they can help you determine which steps to perform to complete the recovery.
37	File system full	Admin error	An attempt to write a file failed because there was insufficient space on the file system. SnapDrive for UNIX can proceed when you free enough space on the appropriate file system.
38	File error	Admin error	An I/O error occurred when SnapDrive for UNIX was reading or writing a system configuration file or a temporary file.
39	Duplicate diskgroup	Command error	SnapDrive for UNIX got a duplicate minor node number when trying to activate a disk group.
40	File system thaw failed.	Admin error	A snap create command failed due to system activity on the file system. This usually occurs when the SnapDrive file system freeze, required for the snapshot, times out before the snapshot is complete.

Exit value	Error name	Type	Description
41	System configuration query error	Admin error	SnapDrive for UNIX encountered a problem accessing the Object Data Manager (AIX systems only). This can be caused by: ◆ An incorrect ODMPATH environment variable. ◆ An ODM lock, open and held by a crashed or hung process ◆ Corrupt ODM files
43	Name already in use	Command error	SnapDrive for UNIX attempted to create a disk group, host volume, file system or LUN but the name was already in use. To correct, select a name that is not in use, and re-enter the SnapDrive command.
44	File system manager error	Fatal	SnapDrive for UNIX encountered an unexpected error from the file system when: ◆ attempting to create the file system ◆ making an entry in the file system mount table to automatically mount the file system at boot. The text of the error message displayed with this code describes the error that the file system encountered. Record the message, and send it to technical support so that they can help you determine which steps to perform to complete the recovery.
45	Mount point error	Admin error	The file system mount point appeared in the system mount table file. To correct, select a mount point that is not in use or listed in the mount table, and re-enter the SnapDrive for UNIX command.

Exit value	Error name	Type	Description
46	LUN not found	Command error	A SnapDrive for UNIX command attempted to access a LUN that did not exist on the storage system. To correct, check that the LUN exists and that the name of the LUN is entered correctly.
47	Initiator group not found	Admin error	A storage system initiator group could not be accessed as expected. As a result, SnapDrive for UNIX cannot complete the current operation. The specific error message describes the problem and the steps you need to perform to resolve it. Fix the problem and then repeat the command.
48	Object offline	Admin error	Snapdrive for UNIX attempted to access an object (such as a volume) but failed because the object was offline.
49	Conflicting entity	Command error	SnapDrive for UNIX attempted to create an igroup, but encountered an igroup of the same name.
50	Cleanup error	Fatal	Snapdrive for UNIX encountered an item that should be removed but is still there.
51	Disk group ID conflict	Command error	A `snap connect` command requested a disk group ID that conflicts with an existing disk group. This usually means that a `snap connect` originating host operation is being attempted on a system that does not support it. To fix this problem, attempt the operation from a different host.
52	LUN not mapped to any host	Admin error	A LUN is not mapped to any host. In other words, it does not belong to a storage system initiator group. To be accessible, the LUN must be mapped to the current host outside SnapDrive for UNIX.

Exit value	Error name	Type	Description
53	LUN not mapped to local host	Admin error	A LUN is not mapped to the current host. In other words, it does not belong to a storage system initiator group that includes initiators from the current host. To be accessible, the LUN must be mapped to the current host outside SnapDrive for UNIX.
54	LUN is mapped using foreign igroup	Admin error	A LUN is mapped using a foreign storage system initiator group. In other words, it belongs to a storage system igroup containing only initiators not found on the local host. As a result, SnapDrive for UNIX cannot delete the LUN. To use SnapDrive for UNIX to delete a LUN, the LUN must belong only to local igroups; that is, igroups containing only initiators found on the local host.
55	LUN is mapped using mixed igroup	Admin error	A LUN is mapped using a mixed storage system initiator group. In other words, it belongs to a storage system igroup containing both initiators found on the local host and initiators not found there. As a result, SnapDrive for UNIX cannot disconnect the LUN. To use SnapDrive for UNIX to disconnect a LUN, the LUN must belong only to local igroups or foreign igroups; not mixed igroups. (Local igroups contain only initiators found on the local host; foreign igroups contain initiators not found on the local host.)

Exit value	Error name	Type	Description
56	Snapshot restore failed	Admin error	Snapdrive for UNIX attempted a `snap restore` operation, but it failed without restoring any LUNs in the snapshot. The specific error message describes the problem and the steps you need to perform to resolve it. Fix the problem and then repeat the command.
58	Host reboot needed	Admin error	The host operating system requires a reboot in order to update internal data. SnapDrive for UNIX has prepared the host for this update, but cannot complete the current operation. Reboot the host and then re-enter the SnapDrive for UNIX command line that caused this message to appear. After the reboot, the operation will be able to complete.

Exit value	Error name	Type	Description
59	Host, LUN preparation needed	Admin error	The host operating system requires an update to internal data in order to complete the current operation. This update is required to allow a new LUN to be created. SnapDrive for UNIX cannot perform the update, because automatic host preparation for provisioning has been disabled because the snapdrive.conf variable `enable-implicit-host-preparation` is set to off. (For more information, see "Setting configuration information" on page 64.) With automatic host preparation disabled, you should use either the `snapdrive config prepare luns` command to prepare the host to provision LUNs or perform the preparation steps manually. To avoid this error message, set the `enable-implicit-host-preparation` value to On in the snapdrive.conf file. **Note** Currently, only the Linux and Solaris platforms requires host preparation.
61	Cannot support persistent mount	Command error	For Linux hosts only, an error occurred because a `snapshot connect` command, `storage connect` command, or `host connect` command requested a persistent mount for a file system that is not on a partitioned LUN.

Exit value	Error name	Type	Description
62	Not empty	Command error	An error occurred because SnapDrive for UNIX could not remove a storage system volume or directory. This may happen when another user or another process creates a file at exact the same time and in the same directory SnapDrive tries to delete. To avoid this error make sure that only one user works with the storage system volume at the time.
63	Timeout expired	Command error	An error occurred because SnapDrive for UNIX could not restore a LUN within the timeout period of 50 minutes. Record the message, and send it to technical support so that they can help you determine which steps to perform to complete the recovery.
64	Service not running	Admin error	An error occurred because a SnapDrive command specified an NFS entity and the storage system was not running the NFS service.
126	Unknown error	Admin error	An unknown error occurred that might be serious. Run the snapdrive.dc utility and send its results to technical support for analysis.
127	Internal error	Fatal	A SnapDrive for UNIX internal error occurred. Run snapdrive.dc and send its results to technical support for analysis.

Checklists for Using SnapDrive Commands 8

About this chapter

This chapter provides details about creating and using SnapDrive for UNIX snapshots.

Topics in this chapter

This chapter discusses the following topics:

- "Collecting information needed by SnapDrive for UNIX Commands" on page 246
- "Checklist for using the snapshot create command" on page 247
- "Checklist for using the snapshot rename command" on page 250
- "Checklist for using the snapshot show command" on page 252
- "Checklist for using the snapshot restore command" on page 254
- "Checklist for using the snapshot delete command" on page 257
- "Checklist for using the snapshot connect command" on page 259
- "Checklist for using the snapshot disconnect command" on page 263
- "Checklist for using the storage create command" on page 265
- "Checklist for using the storage resize command" on page 269
- "Checklist for using the storage show command" on page 271
- "Checklist for using the storage connect command" on page 275
- "Checklist for using the storage disconnect command" on page 277
- "Checklist for using the host disconnect command" on page 281
- "Checklist for using the storage delete command" on page 284

Collecting information needed by SnapDrive for UNIX Commands

Collecting information needed by commands

This chapter provides checklists you can use as you execute Snapdrive for UNIX. For each command, it supplies the following:

- Recommended usage formats
- Information on the keywords, options, and arguments available with the commands and the values you should supply
- Examples of the commands

These checklists provide a quick overview of using these commands. For more information on the commands, see the chapter "Creating and Using Snapshots" on page 117 and the chapter "Provisioning and Managing Storage" on page 169.

General notes about the commands

The following are general notes about the commands:

- The `-dg` and `-vg` options in the command lines are synonyms that reflect the fact that some operating systems refer to disk groups and others refer to volume groups. This guide uses `-dg` to refer to both disk groups and volume groups
- The `-lvol` and `-hostvol` options in the command lines are synonyms that reflect the fact that some operating systems refer to logical volumes and others refer to host volumes. This guide uses `-hostvol` to refer to both logical volumes and host volumes.
- It is strongly recommended that you use the default igroup and not specify an igroup explicitly by including the `-igroup` option on your command line. For information on how SnapDrive for UNIX handles igroup names, see "Command-line keywords" on page 296 in the "Command Reference" chapter. If you need to specify an igroup, see the SnapDrive for UNIX man page for information on doing that.

Checklist for using the snapshot create command

Checklist for creating snapshots

This checklist is a planning aid for creating snapshots of one or more storage system volumes.

Usage format for snapdrive snap create:

```
snapdrive snap create
{-lun | -dg | -fs | -hostvol } file_spec  [file_spec ...]
[ {-lun |-dg | -fs | -hostvol } file_spec [file_spec...]]
-snapname snap_name [ -force [-noprompt]][-unrelated]
```

The following is a summary of the information you need to supply when you use the `snapdrive snap create` command.

Requirement	Argument	Value you want to supply
Determine which type of storage entity that you want to capture in the snapshot. You can specify NFS entities, LUNs, file systems created directly on LUNs, and LVM entities on a single command line. Supply that entity's name with the appropriate argument. This is the value for the *file_spec* argument. ◆ If you specify a disk group that has a host volume or file specification, the argument translates into a set of disk groups on the storage system. SnapDrive creates the entire disk group containing the entity, even if the entity is a host volume or file system. ◆ If you specify a file specification that is an NFS mount point, the argument translates to the directory tree on the storage system volume. ◆ If you specify a LUN, or a LUN that has a file system, the argument translates to the LUN, or to the LUN that has the file system. Refer to "Guidelines for snapshot creation" on page 122 for information about how SnapDrive freezes data for the host and storage system you specify.		
◆ LUN (`-lun file_spec`)	*name of the LUN. You must include the name of the storage system, volume and LUN.*	
◆ Disk group (`-dg file_spec`) or volume group (`-vg file_spec`)	*name of the disk or volume group*	
◆ File system (`-fs file_spec`)	*filesystem_name*	

Requirement	Argument	Value you want to supply
◆ Host volume (`-hostvol file_spec`) or logical volume (`-lvol file_spec`)	*name of the host or logical volume* **Note** You must supply both the requested volume and the disk group containing it; for example, `-hostvol dg3/acct_volume`.	
Specify the name for the snapshot. This can be either the long version of the name that includes the storage system and volume with the snapshot name or the short version that is just the snapshot name.		
◆ Snapshot name (`-snapname snap_name`)	*snapshot_name*	
◆ (**Optional**) Decide if you want to create a snapshot of storage entities that have no dependent writes when the snapshot is taken. Because the entities have no dependent writes, SnapDrive creates a crash-consistent snapshot of the individual storage entities, but does not take steps to make the entities consistent with each other		
◆ `-unrelated`	~	
(**Optional**) Decide if you want to overwrite an existing snapshot. Without this option, this operation halts if you supply the name of an existing snapshot. When you supply this option and specify the name of an existing snapshot, the command prompts you to confirm that you want to overwrite the snapshot. To prevent SnapDrive for UNIX from displaying the prompt, include the `-noprompt` option also. (You must always include the `-force` option if you want to use the `-noprompt` option.)		
◆ `-force`	~	
◆ `-noprompt`	~	

Examples of snap create command lines

The following are some sample command lines that use the `snapdrive snap create` command.

```
# snapdrive snap create -dg -dg_name1 dg_name2 -snapname snap_name
# snapdrive snap create -fs /fs_mount_dir -lvol dg1/logical_volname
  -snapname snap1
# snapdrive snap create -fs /mnt/fs1 -snapname snapfs1
# snapdrive snap create -fs /mnt/fs1 /mnt/fs2 -snapname fs12
```

```
# snapdrive snap create -lun ham:/vol/vol1/luna lunb
-snapname twoluns

# snapdrive snap create -lun ham:/vol/vol1/luna eggs:/vol/vol1/lunb
-snapname lunab
```

Checklist for using the snapshot rename command

Checklist for renaming snapshots

This checklist is a planning aid for renaming snapshots.

Usage format for snapdrive snap rename:

```
snapdrive snap rename [-snapname] old_long_snap_name new_snap_name
  [-force [-noprompt]]
```

Note

An error occurs if you try to rename a snapshot to a different storage system volume.

An error occurs if the new name for the snapshot already exists. You can use the `-force` option to force SnapDrive for UNIX to change the name without generating an error.

The following is a summary of the information you need to supply when you use the `snapdrive snap rename` command.

Requirement	Argument	Value you want to supply
Enter the `-snapname` option followed by the name of the existing snapshot and then the name to which you want to change it. You must use the long form of the snapshot name for the *old_long_snap_name*, where you enter the storage system name, volume, and snapshot name. You can use the short form of the name for the *new_snap_name* because it must exist on the same storage system volume as the original snapshot.		
◆ `-snapname`		
◆ Current name of the snapshot; use the long form of this name	*old_long_snap_name*	
◆ New name of the snapshot	*new_snap_name*	
(**Optional**) Decide if you want to overwrite an existing snapshot. Without this option, this operation halts if you supply the name of an existing snapshot. When you supply this option and specify the name of an existing snapshot, it prompts you to confirm that you want to overwrite the snapshot. To prevent SnapDrive for UNIX from displaying the prompt, include the `-noprompt` option also. (You must always include the `-force` option if you want to use the `-noprompt` option.)		
◆ `-force`	~	

Requirement	Argument	Value you want to supply
◆ -noprompt	~	

Examples of snap rename command lines

The following are some sample command lines that use the `snapdrive snap rename` command.

```
# snapdrive snap rename -force -snapname filer1:/vol/vol1:oldsnap
newsnapshot
```

```
# snapdrive snap rename -snapname filer1:/vol/vol1:FridaySnap
filer1:/vol/vol1:Snap040130
```

Checklist for using the snapshot show command

Checklist for displaying snapshots

This checklist is a planning aid for displaying information about snapshots.

Note

You can use either `snapdrive snap show` or `snapdrive snap list` in these command lines. These commands are synonyms. The usage formats use `snapdrive snap show` to make this guide easier to read.

Usage formats for snapdrive snap show:

Display information about snapshots residing on a storage system:

```
snapdrive snap show -filer filername [filername...]
  [-verbose]
```

Display information about snapshots of a storage system volume:

```
snapdrive snap show -filervol filervol [filervol...]
  [-verbose]
```

Display information about snapshots of a LUN, disk or volume group, file system, or host or logical volume:

```
snapdrive snap { show | list } {-lun |-dg | -fs | -hostvol }
file_spec [file_spec ...]   [-verbose]
```

Display information about a snapshot:

```
snapdrive snap show [-snapname] long_snap_name   [snap_name ...]
  [-verbose]
```

Note

The `show` and `list` forms of this command are synonymous.

For SnapDrive for UNIX 2.0, you must use the long form of the snapshot name when you display information about snapshots.

The following is a summary of the information you need to supply when you use the `snapdrive show | list` command.

Requirement	Argument	Value you want to supply
Based on the command you enter, you can display information about any of the following: ◆ Storage systems ◆ Storage system volumes ◆ Disk or volume groups) ◆ File systems ◆ Host or logical volumes ◆ Snapshots The value you enter for the *file_spec* argument must identify the storage entity about which you want to display information. The command assumes the entities are on the current host.		
◆ Storage system (`-filer`)	*filername*	
◆ A volume on the storage system (`-filervol`)	*filervol*	
◆ Disk group (`-dg file_spec`) or volume group (`-vg file_spec`)	*name of the disk or volume group*	
◆ File system (`-fs file_spec`)	*filesystem_name*	
◆ Host volume (`-hostvol file_spec`) or logical volume (`-lvol file_spec`)	*name of the host or logical volume*	
If you want to display information about a snapshot, specify the name for the snapshot. For the first snapshot, *long_snap_name*, enter the long version of the name, which includes the storage system name, volume, and snapshot name. You can use the short version of the snapshot name if it is on the same storage system and volume.		
◆ Snapshot name (`-snapname long_snap_name`)	*long_snap_name*	
◆ Additional snapshot names	*snap_name* (long or short version)	
To display additional information, include the `-verbose` option.		
`-verbose`	~	

Checklist for using the snapshot restore command

Checklist for restoring snapshots

This checklist is a planning aid for restoring snapshots.

Usage format for snapdrive snap restore:

```
snapdrive snap restore -snapname snap_name
{ -lun | -dg | -fs | -hostvol | -file } file_spec [file_spec ...]
[{ -lun | -dg | -fs | -hostvol | -file } file_spec [file_spec ...]
...] [-force [-noprompt]] [{-reserve | -noreserve}]
```

The following is a summary of the information you need to supply when you use the `snapdrive snap restore` command.

Requirement	Argument	Value you want to supply
Decide the type of storage entity that you wish to restore and enter that entity's name with the appropriate argument. ◆ If you specify a host volume or file system to be restored, the argument you give is translated to the disk group containing it. SnapDrive then restores the entire disk group. SnapDrive for UNIX freezes any file systems in host volumes in those disk groups and takes a snapshot of all storage system volumes containing LUNs in those disk groups. ◆ If you specify a file specification that is an NFS mount point, the argument translates to a directory tree. SnapDrive for UNIX restores only the NFS directory tree or file. Within the directory tree, SnapDrive will delete any new NFS files or directories that you created after you created the snapshot. This ensures that the state of the restored directory tree will be what it was at the point when the snapshot of the tree was taken. ◆ If you restore a LUN, SnapDrive for UNIX restores the LUN you specify. ◆ If you restore a file system that is created directly on a LUN, SnapDrive for UNIX restores the LUN and the file system.		
◆ LUN (`-lun file_spec`)	*name of the LUN. You must include the name of the storage system, volume and LUN.*	
◆ Disk group (`-dg file_spec`) or volume group (`-vg file_spec`)	*name of the disk or volume group*	
◆ File system (`-fs file_spec`)	*filesystem_name*	
◆ File (`-file file_spec`)	*name of the NFS file*	

Requirement	Argument	Value you want to supply
◆ Host volume (`-hostvol file_spec`) or logical volume (`-lvol file_spec`)	*name of the host or logical volume* **Note** You must supply both the requested volume and the disk group containing it; for example, `-hostvol dg3/acct_volume`.	

Specify the name for the snapshot. If any of the *file_spec* arguments you supply on the command line currently exist on the local host, you can use a short form of the snapshot name.
If none of the *file_spec* arguments exist on the host, you must use a long form of the snapshot name where you enter the storage system name, volume, and snapshot name. The following is an example of a long snapshot name:
`big_filer:/vol/account_vol:snap_20031115`
The short form of the same snapshot name would omit the storage system and storage system volume name, so it would appear as:
`snap_20031115`		

◆ Snapshot name (`-snapname`)	*snapshot_name*	
(**Optional**) Decide if you want SnapDrive to create a space reservation when you restore the snapshot.		
`-reserve | -noreserve`		
(**Optional**) Decide if you want to overwrite an existing snapshot. Without this option, this operation halts if you supply the name of an existing snapshot. When you supply this option and specify the name of an existing snapshot, it prompts you to confirm that you want to overwrite the snapshot. To prevent SnapDrive for UNIX from displaying the prompt, include the `-noprompt` option also. (You must always include the `-force` option if you want to use the `-noprompt` option.)		
◆ `-force`	~	
◆ `-noprompt`	~	

Examples of snap restore command lines

The following are some sample command lines that use the `snapdrive snap restore` command.

```
# snapdrive snap restore -vg volgrp1 volgrp2 -snapname
  filer:/vol/volX:snap_vg1_vg2

# snapdrive snap restore -fs /mnt/fs1 /mnt/fs2 -snapname fs12

# snapdrive snap restore -hostvol dg1/hostvol1 -fs /fs_mnt_dir
  -snapname filer:/vol/volX:snap_name

# snapdrive snap restore -lun filer2:/vol/vol1/luna lunb
-snapname Sunday
```

Checklist for using the snapshot delete command

Checklist for deleting snapshots

This checklist is a planning aid for deleting snapshots.

Usage format for snapdrive snap delete:

```
snapdrive snap delete
  [-snapname} long_snap_name [snap_name...]
  [-verbose] [-force [-noprompt]]
```

Note

If the snapshot you specify is in use, this operation fails. SnapDrive for UNIX only reports that this operation completed successfully if all the snapshots are removed.

The following is a summary of the information you need to supply when you use the `snapdrive snap delete` command.

Requirement	Argument	Value you want to supply
Specify the name for the snapshot. Use the long form of the snapshot name where you enter the storage system name, volume, and snapshot name. The following is an example of a long snapshot name: `big_filer:/vol/account_vol:snap_20031115` If you want to specify additional snapshots, you can use the short form of the name if they are on the same storage system and volume as the first snapshot. Otherwise, use the long form of the name again.		
◆ Snapshot name (`-snapname`)	*long_snapshot_name*	
◆ Additional snapshots	*snapshot_name* (either long or short form)	
To display a list of the snapshots being deleted, include the -verbose option. This option fills in the missing storage system and volume information in cases where you used the short form of the snapshot name.		
◆ `-verbose`	~	

Example of snap delete command lines

The following command line uses the `snapdrive snap delete` command.

```
# snapdrive snap delete -v filer1:/vol/vol0/snap1 snap2 snap3
       snapdrive: deleting
               filer1:/vol/vol0/snap1
               filer1:/vol/vol0/snap2
               filer1:/vol/vol0/snap3
```

Checklist for using the snapshot connect command

Checklist for connecting snapshots

This checklist is a planning aid for connecting snapshots. It summarizes the information you need to provide.

Usage format for snapdrive snap connect:

Connect a snapshot that contains LUNs:

```
snapdrive snap connect -lun s_lun_name d_lun_name
[[-lun] s_lun_name d_lun_name ...] -snapname long_snap_name
[-igroup ig_name [ig_name ...]]
```

Connect a snapshot that contains storage entities:

```
snapdrive snap connect fspec_set [fspec_set...]
-snapname long_snap_name [-igroup ig_name [ig_name ...]]
[-autoexpand] [-autorename] [-nopersist] [{-reserve | -noreserve}]
[-readonly]
```

where *fspec_set* has the format:

```
  {-dg | -fs | -hostvol} src_file_spec   [dest_file_spec]
[{-destdg | -destvg} dgname]    [{-destlv | -desthv} lvname]
```

Note

This command must always start with the name of the storage entity you want to connect (for example, `-dg`, `-hostvol`, or `-fs`). If you specify an NFS mount point, you cannot specify non-NFS entities (`-vg`, `-dg`, `-fs`, `-lvol` or `-hostvol`) on the same command line.

Any destination names you supply must not currently be in use, or the command fails.

When you use the `-autoexpand` option without the `-autorename` option, the `snap connect` command fails if the default name for the destination copy of an LVM entity is in use. If you include the `-autorename` option, SnapDrive for UNIX renames the entity when the default name is in use. This means that with the `-autorename` option on the command line, the snap connect continues regardless of whether all the necessary names are available.

If you do not include `-autoexpand` and you do not specify all the LVM host volumes in all the disk groups referred to on the command line, either by specifying the host volume itself or the file system), the command fails.

The following is a summary of the information you need to supply when you use the `snapdrive snap connect` command.

Requirement	Argument	Value you want to supply
Decide the type of storage entity that you want to use to attach the snapshot and supply that entity's name with the appropriate argument. This is the value for the *src_fspec* argument. ◆ If you connect a snapshot of a LUN, SnapDrive for UNIX connects the LUN you specify. You can not use the `-lun` option on the same command line with the `-vg -dg -fs -lvol` or `-hostvol` options. ◆ If you connect a snapshot of a file system that is created directly on a LUN, SnapDrive connects the LUN that has the file system. ◆ If you connect a snapshot of a disk group that has a host volume or file specification, the argument translates into a set of disk groups on the storage system. SnapDrive connects the entire disk group containing the entity, even if the entity is a host volume or file system. ◆ If you connect a snapshot of an NFS file system, the argument translates to the NFS directory tree. SnapDrive creates a FlexClone of the volume, removes directory trees that are not specified in the snapshot, and then connects and mounts the NFS directory tree. If you specify an NFS mount point, you cannot specify non-NFS entities (`-vg`,`-dg`, `-fs`, `-lvol` or `-hostvol`) on the same command line.		
◆ LUN (`-lun` *`file_spec`*)	*short name of the LUN.*	
◆ Disk group (`-dg` *`file_spec`*) or volume group (`-vg` *`file_spec`*)	*name of the disk or volume group*	
◆ File system (`-fs` *`file_spec`*)	*filesystem_name*	
◆ Host volume (`-hostvol` *`file_spec`*) or logical volume (`-lvol` *`file_spec`*)	*name of the host or logical volume*	
Connect a snapshot with an NFS directory tree to Data ONTAP 6.5 or 7.0 configurations. ❖ If your configuration uses Data ONTAP 6.5, or a later version of Data ONTAP with traditional (not FlexVol) volumes, you must specify this option to connect the snapshot with read only access (required). ❖ If your configuration uses Data ONTAP 7.0 and FlexVol volumes, SnapDrive for UNIX automatically provides read-write access. Specify this option only if you want to restrict access to read only (optional).		
◆ `-readonly`		

Requirement	Argument	Value you want to supply
(**Optional**) Supply a name by which the target entity will be accessible after the storage entity is connected. SnapDrive for UNIX uses this name to connect the destination entity. This is the *dest_file_spec* argument. If you omit this name, the `snap connect` command uses the value you supplied for *src_fspec*.		
◆ Name of target entity	*dest_file_spec*	
(**Optional**) Specify the names for the destination storage entities. If you included this information as part of the *dest_fspec*/*src_fspec* pair, you do not need to enter it here. If you do not specify the name needed to connect an entity in the destination disk group, the `snap connect` command takes the name from the source disk group. If it cannot use that name, the operation fails, unless you included -autorename on the command line.		
◆ Destination disk group (`-destdg`) or destination volume group (`-destvg`)	*dgname*	
◆ Destination logical volume (`-destlv`) or destination host volume (`-desthv`)	*lvname*	
Specify the name for the snapshot. Use the long form of the name where you enter the storage system name, volume, and snapshot name.		
◆ Snapshot name (`-snapname`)	*long_snap_name*	
(**Optional**) Connect the snapshot to a new location without creating an entry in the host file system table		
◆ `-nopersist`	~	
(**Optional**) Connect the snapshot to a new location with or without creating a space reservation		
◆ -reserve | -noreserve	~	
(**Optional:** It is recommended that you use the default igroup for your host instead of supplying an igroup name.)		
◆ Igroup name (`-igroup`)	*ig_name*	

Requirement	Argument	Value you want to supply
To shorten the amount of information you must supply when connecting to a volume group, include the `-autoexpand` option on the command line. This option lets you name only a subset of the logical volumes or file systems in the volume group. It then expands the connection to the rest of the logical volumes or file systems in the disk group. This way you do not need to specify each logical volume or file system. SnapDrive for UNIX uses this information to generate the name of the destination entity. This option applies to each disk group specified on the command line and all host LVM entities within the group. It is recommended that, if you include this option, you should also include the `-autorename` option. If the `-autoexpand` option needs to connect the destination copy of an LVM entity, but the name is already in use, the snap connect command fails unless the `-autorename` option is on the command line.		
`-autoexpand`	~	
The `-autorename` option enables the `snap connect` command to rename a destination LVM entity if the default name set by the `-autoexpand` option is already in use. This option applies to all host-side entities specified on the command line. If you include the `-autorename` option on the command line, it implies the `-autoexpand` option, even if you don't include that option.		
`-autorename`	~	

Examples of snap connect command lines

The following are some sample command lines that use the `snapdrive snap connect` command.

```
# snapdrive snap connect -hostvol srcdg/myvol   destdg/myvol
  -snapname filer1:/vol/vol1:mysnapshot
```

```
# snapdrive snap connect -vg myvg -snapname snap1
```

```
# snapdrive snap connect -dg src_dg1 dest_dg1
  -snapname filer1:/vol/vol1:mysnapshot
```

```
# snapdrive snap connect -lun mylun -snapname
toaster:/vol/vol1:snapmylun
```

```
./snapdrv snap connect -lun mylun1 mylun1copy -snapname
hornet:/vol/vol1:tuesdaysnapshot
```

```
# ./snapdrv snap connect -lun mylun1 mylun1copy -lun mylun2
mylun2copy -snapname hornet:/vol/vol1:tuesdaysnapshot
```

Checklist for using the snapshot disconnect command

Checklist for disconnecting snapshots

This checklist is a planning aid for disconnecting snapshots. It summarizes the information you need to provide.

Usage format for snapdrive snap disconnect:

Disconnect a snapshot that contains LUNS:

```
snapdrive snap disconnect -lun long_lun_name [lun_name ...]
```

Disconnect a snapshot that contains storage entities:

```
snapdrive snap disconnect {-dg | -fs | -hostvol } file_spec
[file_spec ...]   [-dg | -fs | -hostvol } file_spec [file_spec
...]]   [-full]
```

Note

This command must always start with the storage entity (for example, `-lun -dg`, `-hostvol`, or `-fs`).

- If you specify a LUN (`-lun`), you must enter the long LUN name. You cannot specify a LUN with the `-lun` option on the same command line as other storage entities (`-vg`,`-dg`, `-fs`, `-lvol` or `-hostvol` options).
- If you specify an NFS mount point, you cannot specify non-NFS entities (`-vg`, `-dg`, `-fs`, `-lvol` or `-hostvol`) on the same command line. You must use a separate command to specify the NFS mountpoint.

An error occurs if the host entity is using LUNs that are not part of the snapshot.

An error also occurs if you specify a subset of the host volumes and/or file systems contained in each target disk group.

The following is a summary of the information you need to supply when you use the `snapdrive snap disconnect` command.

Requirement	Argument	Value you want to supply
Specify the type of storage entity that you want to use to disconnect the snapshot and supply that entity's name with the appropriate argument. This is the value for the *file_spec* argument.		

Requirement	Argument	Value you want to supply
◆ LUN (`-lun file_spec`)	*name of the LUN. Include the name of the filer, volume and LUN.*	
◆ Disk group (`-dg file_spec`) or volume group (`-vg file_spec`)	*name of the disk or volume group*	
◆ File system (`-fs file_spec`)	*filesystem_name*	
◆ Host volume (`-hostvol file_spec`) or logical volume (`-lvol file_spec`)	*name of the host or logical volume*	
Include the -full option on the command line if you want SnapDrive for UNIX to disconnect the objects from the snapshot even if a host-side entity on the command line has other entities (such as a disk group that has one or more host volumes). If you do not include this option, you must specify only empty host-side entities.		
`-full`	~	

Examples of snap disconnect command lines

The following are some sample command lines using the `snapdrive snap disconnect` command.

```
# snapdrive snap disconnect -hostvol dg5/myvolume

# snapdrive snap disconnect -vg vg1 vg10

# snapdrive snap disconnect -fs mnt/fs1

# snapdrive snap disconnect -lun toaster:/vol/vol1/luna

# snapdrive snap disconnect -quiet -dg dg1 dg2 dg3
```

Checklist for using the storage create command

Checklist for creating storage

This checklist is a planning aid for creating storage. It summarizes the information you need to provide.

Usage format for snapdrive storage create:

Creating LUNs:

```
snapdrive storage create -lun long_lun_name [lun_name ...] -lunsize
size [{-reserve |-noreserve }][-igroup ig_name [ig_name ...]]
```

Creating a file system on a LUN and setting up the LUN automatically

```
snapdrive storage create -fs file_spec -nolvm [-fstype type]
[-fsopts options] [-mntopts options] [-nopersist] -filervol
long_filer_path -lunsize size [-igroup ig_name [ig_name ...]]
[{ -reserve | -noreserve }]
```

Creating a file system on a LUN and specifying the LUN

```
snapdrive storage create -fs file_spec -nolvm [-fstype type] [-
fsopts options] [-mntopts options] [-nopersist] -lun long_lun_name
-lunsize size [-igroup ig_name [ig_name ...]]
[{ -reserve | -noreserve }]
```

Creating LVM entities and having SnapDrive for UNIX automatically set up the underlying LUN:

```
snapdrive storage create host_lvm_fspec -filervol long_filer_path
-dgsize size [-igroup ig_name [ig_name ...]]
[{ -reserve | -noreserve }]
```

The *host_lvm fspec* arguement specifies a filesystem, logical volume or disk group:

```
-fs file_spec [-fstype type] [-fsopts options] [-mntopts options]
[-nopersist] [ -hostvol file_spec] [ -dg dg_name]
```

Creating LVM entities and specifying the LUN.

```
snapdrive storage create host_lvm_fspec -lun long_lun_name
[lun_name ...] -lunsize size [-igroup ig_name [ig_name ...]]
[{ -reserve | -noreserve }]
```

Note

It is not possible to construct a disk group (or, therefore, a logical volume or file system) using LUNs from more than one storage system volume. If you supply multiple volumes in the list of LUNs given to `-lun`, do not include the `-dg` option.

The following is a summary of the information you need to supply when you use the `snapdrive storage create` command.

Requirement	Argument	Value you want to supply
Decide the type of storage you want to provision. Based on the command you enter, you can create any of the following: ◆ LUNs If you create one or more LUNs, the first argument must use the long form of the LUN name, which specifies the storage system name, the volume name, and the name of the LUN within the volume. To specify additional LUNs, you can use the LUN name alone if the new LUN is on the same storage system and volume as the previous LUN. Otherwise, you can specify a new storage system name and volume name (or just a volume name) to replace the previous values. ◆ A file system created directly on a LUN If you create a file system on a LUN, the first argument must be the `-fs` mountpoint. To create the file system on a LUN in a storage system and volume, use the `-filervol` argument and specify the name of the storage system and volume. To create the file system on a specific LUN, use the `-lun` argument and specify the storage system name, volume name, and LUN name. You must also include the `-nolvm` option. ◆ LVM disk groups with host volumes and file systems When you specify a disk or volume group, file system, or host or logical volume, SnapDrive for UNIX performs all the actions necessary to create the entity you specify. You can either explicitly specify the LUNs, or just supply the storage system and volume information and let SnapDrive for UNIX create the LUNs automatically.		
◆ A LUN (`-lun`)	*long_lun_name*	
◆ Additional LUNs	*lun_name* (long or short form)	
◆ Disk group (`-dg` *dgname*) or volume group (`-vg` *vgname*)	*disk or volume group name*	
◆ Host volume (`-hostvol` *file_spec*) or logical volume (`-lvol` *file_spec*)	*Host or logical volume name*	

Requirement	Argument	Value you want to supply
◆ File system (`-fs file_spec`)	*filesystem_name*	
If you are creating a file system that resides directly on a LUN, specify the following argument (required)		
◆ `-nolvm`	~	
Specify the size in bytes of each entity being created.		
◆ Lun size (`-lunsize`)	*size*	
◆ Disk group size (`-dgsize`) ◆ Volume group size (`-vgsize`)	*size*	
Specify the storage system and its volume where you want SnapDrive for UNIX to create the LUNs automatically. ◆ Use the `-filervol` option when you use the command line format that has SnapDrive for UNIX automatically create the LUNs. ◆ Use the -lun option to name the LUNs that you want to use.		
◆ Path to storage system volume (`-filervol`)	*long_filer_path*	
◆ (`-lun`)	*long_lun_path*	
If you are creating a file system, supply the string representing the file system type. The following are the values SnapDrive for UNIX accepts for each host platform: ◆ Solaris: VxFS ◆ HP-UX: VxFS ◆ AIX: JFS2 ◆ Linux: ext3 **Note** By default, SnapDrive for UNIX supplies this value if there is only one file system type for your host platform. In that case, you do not need to enter it.		
◆ File system type (`-fstype`)	*type*	

Requirement	Argument	Value you want to supply
(**Optional**) If you are creating a file system, you can specify the following options: `-fsopts` to specify options you want to pass to the host command used to create the file systems. For example, you might supply options that the `mkfs` command would use. The value you supply usually needs to be specified as a quoted string and must contain exactly the text to be passed to the command. `-mntopts` option to specify options that you want passed to the host mount command (for example, to specify host system logging behavior). `-nopersist` option to create the file system without adding an entry to the file system mount table file on the host. `-reserve | -noreserve` to create the storage with or without creating a space reservation.		
◆ (`-fsopts` *options*)	*option name and value*	
◆ `-nopersist`	~	
◆ `-reserve | -noreserve`	~	
(**Optional:** It is recommended that you use the default igroup for your host instead of supplying an igroup name.)		
◆ Igroup name (`-igroup`)	*ig_name*	

Examples of storage create command lines

The following are some sample command lines that use the `snapdrive storage create` command.

```
# snapdrive storage create -lun kipper:/vol/vol1/lunA lunB lunC
  -dg data_dg -lunsize 10g

# snapdrive storage create -lvol dg5/myvol -filervol
  filer2:/vol/acct -dgsize 500m

# snapdrive storage create -fs /mnt/acct1 -filervol
acctfiler:/vol/vol1 -lunsize 100m -nolvm

# snapdrive storage create -fs /mnt/acct1 -lun
acctfiler:/vol/vol1/luna -lunsize 100m -nolvm
```

Checklist for using the storage resize command

Checklist for resizing storage

This checklist is a planning aid for resizing storage.

Usage format for snapdrive storage resize:

```
snapdrive storage resize
  -dg file_spec [file_spec ...]
  {-growby | -growto} size
  [-addlun [-igroup ig_name [ig_name ...]]] [{ -reseserve | -
noreserve }]]
```

Note

You cannot use the `snapdrive storage resize` command to reduce the size of an entity. You can only increase the size.

The `snapdrive storage resize` command is not supported directly on logical volumes or file systems. For example, you can not use the `snapdrive storage resize` command to resize a file system on a LUN.

The following is a summary of the information you need to supply when you use the `snapdrive storage resize` command.

Requirement	Argument	Value you want to supply
Decide whether you want to increase the size of a disk or volume group and enter that entity's name with the appropriate argument.		
◆ Disk group (`-dg file_spec`) or volume group (`-vg file_spec`)	*name of the disk or volume group*	
Decide how you want to increase the storage size. Keep the following in mind as you use this command: ◆ Use the `-growby` option to increase the entity size by the bytes specified in the *size* argument. ◆ Use the `-growto` option to increase the entity size so that the new total size is the number of bytes specified in the *size* argument. ◆ Use the `-addlun` option to increase the entity size by adding a new, internally-generated LUN to the underlying disk group. If you do not use this argument, SnapDrive for UNIX increases the size of the last LUN in the disk group to meet the byte size specified in either the `-growby` option or the `-growto` option.		

Requirement	Argument	Value you want to supply
◆ Specify the number of bytes by which you want to increase the storage (`-growby size`)	*number_of_bytes*	
◆ Specify the size in bytes that you want the storage to reach (`-growto size`)	*number_of_bytes*	
◆ Tell SnapDrive for UNIX to increase the size by adding a new LUN to the disk group `(-addlun)`	~	
◆ Tell SnapDrive for UNIX to increase the size with or without creating a space reservation `-reserve | -noreserve`		
(**Optional:** It is recommended that you use the default igroup for your host instead of supplying an igroup name.)		
◆ Igroup name (`-igroup`)	*ig_name*	

Examples of storage resize command lines

The following are some sample command lines that use the `snapdrive storage resize` command.

```
# snapdrive storage resize -dg my_dg -addlun -growby 155m
# snapdrive storage resize -dg my_dg -growto 155m
```

Checklist for using the storage show command

Checklist for displaying storage information

This checklist is a planning aid for displaying information about storage.

Usage formats for snapdrive storage show:

Display information about LUNs on a storage system:

```
snapdrive storage { show | list } -filer filer_name
  [filer_name ...]
  [-verbose]
  [-quiet]
```

Display information about LUNs on a storage system volume:

```
snapdrive storage { show | list } -filervol long_filer_path
[filer_path...]  [-verbose]  [-quiet]
```

Display information about a LUN:

```
snapdrive storage { show | list } -lun long_lun_name
  [lun_name ...]   [-verbose]  [-quiet]
```

Display information about an NFS entity, disk or volume group, file system, or host or logical volume:

```
snapdrive storage { show | list } {-dg | -fs | -hostvol } file_spec
  [file_spec ...]   [{-dg | -fs | -hostvol } file_spec
[file_spec ...]]   [-verbose]  [-quiet]
```

Display information about all the devices known to the host:

```
snapdrive storage { show | list } -devices
```

Display information about all the devices and LVM entities known to the host:

```
snapdrive storage { show | list } all
```

Note

The `show` and `list` forms of this command are synonymous.

For SnapDrive for UNIX 2.0, you must use the long form of the snapshot name when you display information about snapshots.

The following is a summary of the information you need to supply when you use the `snapdrive show` command.

Examples of

Requirement	Argument	Value you want to supply
Based on the command you enter, you can display information about any of the following: ◆ LUNs on a storage system accessible to the current host ◆ LUNs on storage system volumes accessible to the current host ◆ Specific LUNs ◆ DIsk or volume groups ◆ File systems ◆ Host or logical volumes The value you enter for the *file_spec* argument must identify the storage entity about which you want to display information. The command assumes the entities are on the current host. When you display information about a LUN on a storage system volume, you must supply the path to the LUN, including the name of the storage system and the volume, as the value for *long_filer_path* and *long_LUN_name.*		
◆ Storage system (`-filer`)	*filer_name*	
◆ LUNs on a storage system volume on the storage system (`-filervol`)	*long_filer_path*	
◆ Specific LUNs (`-lun`)	*long_LUN_name*	
◆ Disk group (`-dg file_spec`) or volume group (`-vg file_spec`)	*name of the disk or volume group*	
◆ File system (`-fs file_spec`)	*filesystem_name*	

Requirement	Argument	Value you want to supply
◆ Host volume (`-hostvol file_spec`) or logical volume (`-lvol file_spec`)	*name of the host or logical volume*	
To display information about all devices known to the host, use the `-devices` option.		
`-devices`	~	
To display information about all devices and LVM entities known to the host, include the `-all` option.		
`-all`	~	
To display additional information, include the `-verbose` option.		
`-verbose`	~	
To prevent all output, both diagnostic and normal output, include the `-quiet` option. This option overrides the `-verbose`, which can be useful for existence checking within a script.		
`-quiet`	~	

storage show command lines

The following are sample `snapdrive storage show | list` command lines.

```
# snapdrive storage show -verbose -filer BigFiler3 BigFiler4

# snapdrive storage show -dg dg1 dg2

# snapdrive storage show -quiet -dg QA_dg_test

# snapdrive storage show -verbose -filervol BigFiler3:/vol/vol1

# snapdrive storage show -lun BigFiler3:/vol/vol1:lunA lunB lunC

# snapdrive storage show -filer BigFiler3 BigFiler4

# snapdrive storage show -fs /mnt/fs21 /mnt/fs41

# snapdrive storage show -hostvol dg2/vol1 dg4/myvol3

# snapdrive storage show -dg dg2 dg4

# snapdrive storage show -dg dg2 -fs /mnt/fs42
```

```
# snapdrive storage show -hostvol dg2/vol3 -fs /mnt/fs42

# snapdrive storage show -devices

# snapdrive storage show -all
```

Checklist for using the storage connect command

Checklist for connecting storage

This checklist is a planning aid for connecting storage. It summarizes the information you need to provide.

Usage format for snapdrive storage connect:

To connect a LUN:

```
snapdrive storage connect -lun long_lun_name [lun_name...]
  [-igroup ig_name [ig_name...]
```

To connect a file system created directly on a LUN:

```
snapdrive storage connect -fs file_spec -nolvm -lun long_lun_name
[-igroup ig_name [ig_name ...]] [-nopersist]
```

To connect LUNs with disk groups, host volumes and file systems:

```
snapdrive storage connect -fs file_spec -hostvol file_spec -lun
long_lun_name [lun_name ...] [-igroup ig_name [ig_name ...]]
[-nopersist]
```

The following is a summary of the information you need to supply when you use the `snapdrive storage connect` command.

Requirement	Argument	Value you want to supply
Specify the LUNs, the file system created directly on a LUN, or the LVM entity that you want to connect to the host. ◆ If you connect one or more LUNs, the first argument must use the long form of the LUN name, which specifies the storage system name, the volume name, and the name of the LUN within the volume. To specify additional LUNs, you can use the LUN name alone if the new LUN is on the same storage system and volume as the previous LUN. Otherwise, you can specify a new storage system name and volume name (or just a volume name) to replace the previous values. ◆ If you connect a file system created directly on a LUN, you must include the long form of the LUN name, and also the `-nolvm` option. ◆ If you connect a LUN with a disk group, host volume and file system, you must use the `-fs` and `-hostvol` options to specify the file system and host volume. The host volume must include the name of the disk group		
◆ A LUN (`-lun`)	*long_lun_name*	

Requirement	Argument	Value you want to supply
◆ Additional LUNs	*lun_name* (long or short form)	
A file system (`-fs`)	*file_system_name*	
To connect a file system that is created on a LUN		
◆ `-nolvm`		
To connect a file system on a host volume		
◆ Host volume (`-hostvol`)	*disk_group_name* and *host_volume_name*	
(**Optional**) Use the -nopersist option to connect the storage to a new location without creating an entry in the host file system table		
◆ `-nopersist`	~	
(**Optional:** It is recommended that you use the default igroup for your host instead of supplying an igroup name.)		
◆ Igroup name (`-igroup`)	*ig_name*	

Examples of storage connect command lines

The following are sample command lines that use the `snapdrive storage connect` command.

```
# snapdrive storage connect -lun filer1:/vol/vol1/toaster_lun2

# snapdrive storage connect -lun filer1:/vol/vol1/toaster_lun2
  toaster_lun3 toaster_lun4

# snapdrive storage connect -fs /acct/acctfs -lun
toaster:/vol/vol1/acctlun1-nolvm


snapdrive storage connect -fs /acct/acctfs -hostvol
dg1/saleshostvol/ -lun toaster:/vol/vol1/acctlun1
```

Checklist for using the storage disconnect command

Checklist for disconnecting storage

This checklist is a planning aid for disconnecting storage. It summarizes the information you need to provide.

Usage format for snapdrive storage disconnect:

Disconnecting LUNs from the host:

```
snapdrive storage disconnect -lun long_lun_name [lun_name...]
```

Disconnecting LUNs and storage entities from the host:

```
snapdrive storage disconnect
  {-dg | -fs | hostvol} file_spec [file_spec ...]
  [{-dg | -fs | -hostvol} file_spec
  [file_spec ...] ...]   [-full]
```

The following is a summary of the information you need to supply when you use the `snapdrive storage disconnect` command.

Requirement	Argument	Value you want to supply
Based on the command you enter, you can remove mappings from any of the following: ◆ LUNs If you disconnect one or more LUNs, the first argument must use the long form of the LUN name, which specifies the storage system name, the volume name, and the name of the LUN within the volume. To specify additional LUNs, you can use the LUN name alone if the new LUN is on the same storage system and volume as the previous LUN. Otherwise, you can specify a new storage system name and volume name (or just a volume name) to replace the previous values. ◆ File systems on LUNs ◆ Disk or volume groups ◆ File systems on disk or volume groups ◆ Host or logical volumes The value you enter for the *file_spec* argument must identify the storage entity you're disconnecting.		
◆ A LUN (`-lun`)	*long_lun_name*	
◆ Additional LUNs	*lun_name* (long or short form)	

Requirement	Argument	Value you want to supply
◆ Disk group (`-dg file_spec`) or volume group (`-vg file_spec`)	*name of the disk or volume group*	
◆ File system (`-fs file_spec`)	*filesystem_name*	
◆ Host volume (`-hostvol file_spec`) or logical volume (`-lvol file_spec`)	*name of the host or logical volume*	
If you want SnapDrive for UNIX to disconnect the storage you specify even if you include on the command line a host-side entity that has other entities (such as a disk group that has one or more host volumes), include the -full option on the command line. If you do not include this option, you must specify only empty host-side entities.		
`-full`	~	

Examples of storage disconnect command lines

The following are sample command lines that use the `snapdrive storage disconnect` command

.

```
# snapdrive storage disconnect -lun filer1:/vol/vol1/lun2

# snapdrive storage disconnect -fs /mnt/sales
```

Checklist for using the host connect command

Checklist for connecting storage

This checklist is a planning aid for connecting storage. It summarizes the information you need to provide.

Usage format for snapdrive host connect:

To connect a LUN:

```
snapdrive host connect -lun long_lun_name [lun_name ...]
```

To connect file system created directly on a LUN:

```
snapdrive host connect -fs file_spec -nolvm -lun long_lun_name
[-nopersist]
```

To connect LUNs with a disk group, host volume and file system:

```
snapdrive host connect -fs file_spec -hostvol file_spec -lun
long_lun_name [lun_name][-nopersist]
```

The following is a summary of the information you need to supply when you use the `snapdrive storage connect` command.

Requirement	Argument	Value you want to supply
Specify the LUNs, a file system created directly on a LUN, or a LUN with a disk group, host volume and file system that you want to connect to the host. ◆ If you connect one or more LUNs, the first argument must use the long form of the LUN name, which specifies the storage system name, the volume name, and the name of the LUN within the volume. To specify additional LUNs, you can use the LUN name alone if the new LUN is on the same storage system and volume as the previous LUN. Otherwise, you can specify a new storage system name and volume name (or just a volume name) to replace the previous values. ◆ If you connect a a file system that is created directly on a LUN, you must include the long form of the LUN name, and also the `-nolvm` option. ◆ If you connect a LUN with a disk group, host volume and file system, you must use the `-fs` and `-hostvol` options to specify the file system and host volume. The host volume must include the name of the disk group		
◆ A LUN (`-lun`)	*long_lun_name*	
◆ Additional LUNs	*lun_name* (long or short form)	

Requirement	Argument	Value you want to supply
A file system (`-fs`)	*file_system_name*	
To add a file system that is created on a LUN		
◆ `-nolvm`		
To add a file system on a host volume		
◆ Host volume (`-hostvol`)	*disk_group_name* and *host_volume_name*	
(**Optional**) Use the -nopersist option to connect the storage to a new location without creating an entry in the host file system table		
◆ `-nopersist`	~	

Examples of host connect command lines

The following are sample command lines that use the `snapdrive host connect` command.

```
# snapdrive host connect -lun acctfiler:/vol/vol1/lunA lunB

# snapdrive host connect -fs /my_fs -lun toaster:/vol/vol1/acctlun1
-nolvm

# snapdrive host connect -fs /acct/acctfs -hostvol
dg1/saleshostvol/ -lun toaster:/vol/vol1/acctlun1
```

Checklist for using the host disconnect command

Checklist for disconnecting storage from the host

This checklist is a planning aid for disconnecting the storage from the existing host. This command applies only to the host side; it does not affect the storage system. Here is a summary of the information you need to provide.

Usage format for snapdrive host disconnect:

To disconnect LUNs from the host:

```
snapdrive host disconnect -lun long_lun_name [lun_name...]
```

To disconnect a file system created directly on a LUN from the host:

```
snapdrive host disconnect -fs file_spec
```

To disconnect LUNs with disk groups, host volumes and file systems

```
snapdrive host disconnect
  {-dg | -fs | -hostvol} file_spec [file_spec ...]
  [{-dg | -fs | -hostvol} file_spec
  [file_spec ...] ...]
  [-full]
```

The following is a summary of the information you need to supply when you use the `snapdrive host disconnect` command.

Requirement	Argument	Value you want to supply
Based on the command you enter, you can disconnect any of the following storage entities without changing the state of the storage system: ◆ LUNs If you disconnect one or more LUNs, the first argument must use the long form of the LUN name, which specifies the storage system name, the volume name, and the name of the LUN within the volume. To specify additional LUNs, you can use the LUN name alone if the new LUN is on the same storage system and volume as the previous LUN. Otherwise, you can specify a new storage system name and volume name (or just a volume name) to replace the previous values.File systems created directly on a LUN ◆ A file system on a LUN ◆ Disk or volume groups ◆ File systems on disk or volume groups ◆ Host or logical volumes The value you enter for the *file_spec* argument must identify the storage entity you're disconnecting.		
◆ Disk group (`-dg file_spec`) or volume group (`-vg file_spec`)	*name of the disk or volume group*	
◆ File system (`-fs file_spec`)	*filesystem_name*	
◆ Host volume (`-hostvol file_spec`) or logical volume (`-lvol file_spec`)	*name of the host or logical volume*	
◆ LUN (`-lun long_lun_name`)	*long LUN name*	
If you want SnapDrive for UNIX to disconnect the storage you specify even if you include on the command line a storage entity that has other entities (such as a disk group that has one or more host volumes), include the -full option on the command line. If you do not include this option, you must specify only empty host-side entities.		
`-full`	~	

Examples of host disconnect commands

The following are some sample command lines that use `snap host disconnect`.

```
snapdrive host disconnect -lun toaster:/vol/vol1/luna lunb

snapdrive host disconnect -lun toaster:/vol/vol1/luna -dg dg1

snapdrive host disconnect -fs mnt:/fs1

snapdrive host disconnect -dg dg1 dg2 dg3
```

Checklist for using the storage delete command

Checklist for deleting storage

This checklist is a planning aid for deleting storage.

Usage format for snapdrive storage delete:

```
snapdrive storage delete -lun long_lun_name [lun_name...]

snapdrive storage delete
  {-dg | -fs | -hostvol} file_spec [file_spec ...]
 [{-dg | -fs | -hostvol} file_spec [file_spec ...] ...]
  [-full]
```

Note

If the storage you specify is in use, this operation fails.

The following is a summary of the information you need to supply when you use the `snapdrive storage delete` command.

Requirement	Argument	Value you want to supply
Based on the command you enter, you can delete any of the following types of storage: ◆ LUNs Specify one or more LUNs that you want to delete from the storage system. If you delete one or more LUNs, the first argument must use the long form of the LUN name, which specifies the storage system name, the volume name, and the name of the LUN within the volume. To specify additional LUNs, you can use the LUN name alone if the new LUN is on the same storage system and volume as the previous LUN. Otherwise, you can specify a new storage system name and volume name (or just a volume name) to replace the previous values. ◆ A file system created directly on a LUNs ◆ Disk or volume groups ◆ File systems on disk or volume groups ◆ Host or logical volumes The value you enter for the *file_spec* argument must identify the storage entity you're deleting.		
◆ A LUN (`-lun`)	*long_lun_name*	
◆ Additional LUNs	*lun_name* (long or short form)	

Requirement	Argument	Value you want to supply
◆ Disk group (`-dg file_spec`) or volume group (`-vg file_spec`)	*name of the disk group or volume group*	
◆ File system (`-fs file_spec`)	*filesystem_name*	
◆ Host volume (`-hostvol file_spec`) or logical volume (`-lvol file_spec`)	*name of the host volume or logical volume* **Note** You must supply both the requested volume and the disk group containing it; for example, `-hostvol dg3/acct_volume`.	
If you want SnapDrive for UNIX to delete the storage you specify even if you include on the command line a host-side entity that has other entities (such as a disk group that has one or more host volumes), include the -full option on the command line. If you do not include this option, you must specify only empty host-side entities.		
`-full`	~	

Command Reference 9

About this chapter

This chapter contains a list of the commands supported in SnapDrive for UNIX and information about the keywords and arguments that work with them.

Topics in this chapter

This chapter discusses the following topics:

- "Summary of the SnapDrive for UNIX commands" on page 288
- "SnapDrive for UNIX keywords and arguments" on page 292

Summary of the SnapDrive for UNIX commands

Command summary

SnapDrive for UNIX supports the following command lines:

```
snapdrive config access {show | list} filer-name

snapdrive config check luns

snapdrive config delete filername [filername ...]

snapdrive config list

snapdrive config prepare luns -count count

snapdrive config set user_name filername [filername ...]

snapdrive config show [host_file_name]

snapdrive host connect -lun long_lun_name [lun_name ...]

snapdrive host connect -fs file_spec -nolvm -lun long_lun_name
[-nopersist]

snapdrive host connect -fs file_spec -hostvol file_spec -lun
long_lun_name [lun_name][-nopersist]

snapdrive host disconnect -lun long_lun_name [lun_name...]

snapdrive host disconnect -fs file_spec

snapdrive host disconnect {-vg | -dg | -fs | -lvol | -hostvol}
file_spec [file_spec ...] [{-vg | -dg | -fs | -lvol | -hostvol}
file_spec [file_spec ...] ...] [-full]

snapdrive snap connect -lun s_lun_name d_lun_name
[[-lun] s_lun_name d_lun_name ...] -snapname long_snap_name
[-igroup ig_name [ig_name ...]]

snapdrive snap connect fspec_set [fspec_set...] -snapname
long_snap_name [-igroup ig_name [ig_name ...]] [-autoexpand]
[-autorename] [-nopersist] [{-reserve | -noreserve}] [-readonly]
```

Note

The *fspec_set* argument has the following format:
{-lun | -vg | -dg | -fs | -lvol | -hostvol} *src_fspec* [*dest_fspec*]
{-destdg | -destvg } dg_name] [{-destlv | -desthv } *lv_name*]

```
snapdrive snap create { -lun | -vg | -dg | -fs | -lvol | -hostvol }
file_spec [file_spec ...] [ { -lun | -vg | -dg | -fs | -lvol |
-hostvol } file_spec [file_spec...]] -snapname snap_name
[-force [-noprompt]] [-unrelated]
```

```
snapdrive snap delete [-snapname} long_snap_name [snap_name...]
[-verbose] [-force [-noprompt]]

snapdrive snap disconnect -lun long_lun_name [lun_name ...]

snapdrive snap disconnect {-vg | -dg | -fs | -lvol | -hostvol}
file_spec [file_spec ...] [-vg | -dg | -fs | -lvol | -hostvol}
file_spec [file_spec ...]] [-full]

snapdrive snap rename [-snapname] old_long_snap_name new_snap_name
[-force [-noprompt]]

snapdrive snap restore -snapname snap_name
{ -lun | -vg | -dg | -fs | -lvol | -hostvol | -file }
file_spec [file_spec ...]
[{ -lun | -vg | -dg | -fs | -lvol | -hostvol | -file }
file_spec [file_spec ...] ...] [-force [-noprompt]] [{-reserve |
-noreserve}]

snapdrive snap { show | list } -filer filername [filername...]
[-verbose]

snapdrive snap { show | list } -filervol filervol [filervol...]
[-verbose]

snapdrive snap { show | list } {-lun |-vg | -dg | -fs | -lvol
| -hostvol } file_spec [file_spec ...] [-verbose]

snapdrive snap { show | list } [-snapname] long_snap_name
[snap_name ...] [-verbose]

snapdrive storage connect -fs file_spec -nolvm -lun long_lun_name
[-igroup ig_name [ig_name ...]] [-nopersist]

snapdrive storage connect -fs file_spec -hostvol file_spec -lun
long_lun_name [lun_name ...] [-igroup ig_name [ig_name ...]] [-
nopersist]

snapdrive storage connect -lun long_lun_name [lun_name ...]
[-igroup ig_name [ig_name ...]]

snapdrive storage create -lun long_lun_name [lun_name ...] -lunsize
size [{ -dg | -vg } dg_name] [-igroup ig_name [ig_name ...]]
[{ -reserve | -noreserve }]

snapdrive storage create -fs file_spec -nolvm [-fstype type]
[-fsopts options] [-mntopts options] [-nopersist]
{ -lun long_lun_name | -filervol long_filer_path } -lunsize size
[-igroup ig_name [ig_name ...]] [{ -reserve | -noreserve }]


snapdrive storage create host_lvm_fspec -filervol long_filer_path
-dgsize size [-igroup ig_name [ig_name ...]]
[{ -reserve | -noreserve }]
```

```
snapdrive storage create host_lvm_fspec -lun long_lun_name
[lun_name ...] -lunsize size [-igroup ig_name [ig_name ...]]
[{ -reserve | -noreserve }]
```

Note

You can use one of the following three formats for the `-file_spec` argument, depending on the type of storage you want to create. (Remember that `-dg` is a synonym for `-vg` and `-hostvol` is a synonym for `-lvol`.)

To create a file system directly on a LUN:

```
-fs file_spec [-nolvm -fs type] [-fsops options] [-mntopts
options]
```

To create a file system that uses a disk group or host volume:

```
-fs file_spec [-fstype type] [-fsopts options] [-mntops
options] [-hostvol file_spec] [-dg dg_name]
```

To create a logical or host volume, use this format:

```
[-hostvol file_spec] [-dg dg_name]
```

To create a disk group, use this format:

```
-dg dg_name
```

```
snapdrive storage delete -lun long_lun_name [lun_name ...]
```

```
snapdrive storage delete { -vg | -dg | -fs | -lvol | -hostvol }
file_spec [file_spec ...] [{ -vg | -dg | -fs | -lvol | -hostvol }
file_spec [file_spec ...]] [-full]
```

```
snapdrive storage disconnect -lun long_lun_name [lun_name...]
```

```
snapdrive storage disconnect {-vg | -dg | -fs | -lvol | -hostvol}
file_spec [file_spec ...]   [{-vg | -dg | -fs | -lvol | -hostvol}
file_spec file_spec ...] ...] [-full]
```

```
snapdrive storage resize {-dg | -vg} file_spec [file_spec ...]
{-growby | -growto} size [-addlun [-igroup ig_name [ig_name ...]]]
[{ -reseserve | -noreserve }]]
```

```
snapdrive storage { show | list } -filer filer_name [filer_name
...] [-verbose] [-quiet]
```

```
snapdrive storage { show | list } -filervol long_filer_path
[filer_path...][-verbose][-quiet]
```

```
snapdrive storage { show | list } {-all | device}
```

```
snapdrive storage { show | list } -lun long_lun_name
[lun_name ...] [-verbose] [-quiet]
```

```
snapdrive storage { show | list } { -vg | -dg | -fs | -lvol |
-hostvol } file_spec [file_spec ...] [{ -vg | -dg | -fs | -lvol | -
hostvol } file_spec [file_spec ...]] [-verbose] [-quiet]
```

SnapDrive for UNIX keywords and arguments

Command-line options

SnapDrive for UNIX enables you to include the following options as appropriate with its commands. In certain cases, you can abbreviate these options. For example, you can use `-h` instead of `-help`.

Option	Description
`-addlun`	Tells SnapDrive for UNIX to add a new, internally-generated LUN to a storage entity in order to increase its size.
`-all`	Used with the `storage {show | list}` command to display all devices and LVM entities known to the host
`-autoexpand`	Used with the `snap connect` command to enable you to request that a disk group be connected when you supply a subset of the logical volumes or file systems in the disk group.
`-autorename`	Used with the `snap connect` command to enable the command to rename any newly-connected LVM entities for which the default name is already in use.
`-device`	Used with the `storage {show | list}` command to display all devices known to the host
`-dgsize (synonymous with -vgsize)`	Used with the `storage create` command to specify the size in bytes of the disk group you want to create.
`-force` (or `-f`)	Causes operations to be attempted that SnapDrive for UNIX would not undertake ordinarily. SnapDrive for UNIX prompts you to ask for confirmation before it executes the operation.

Option	Description
`-fsopts`	The options you want passed to the host operation that creates the new file system. Depending on your host operating system, this host operation might be a command such as the `mkfs` command. The argument you supply with this option usually needs to be specified as a quoted string and must contain exactly the text to be passed to the command. For example, you might enter `-o largefiles` as the option you want passed to the host operation.
`-fstype`	The type of file system you are creating. The file system must be a type that SnapDrive for UNIX supports for your operating system. It currently supports the following types: ◆ AIX: JFS2 ◆ HP-UX: VxFS ◆ Linux: Ext3 Current values that you can include with this argument are vxfs HP-UX), jfs2 (AIX), or ext3 (Linux)
`-full`	Allows operations on a specified host-side entity to be performed even if the entity is not empty (for example, the entity might be a volume group containing one or more logical volumes).
`-growby`	The number of bytes you want to add to a LUN or disk group in order to increase its size.

Option	Description
-growto	The target size in bytes for a LUN, disk group, or volume group. SnapDrive for UNIX automatically calculates the number of bytes necessary to reach the target size and increases the size of the object by that number of bytes.
-help (or -h)	Prints out the usage message for this command and operation. Enter this option by itself without other options.
-lunsize	The size of the LUN in bytes to be created by a given command.
-mntopts	Specifies options that you want passed to the host mount command (for example, to specify file system logging behavior). Options are also stored in the host file system table file. Allowable options allowed depend on the host file system type. The -mntopts argument that you supply is a file system-type option that is specified using the mount command "- o" flag. Do not include the "- o" flag in the -mntopts argument.Do not include the "-o" flag in the -mntopts argument. For example, the sequence -mntopts tmplog passes the string -o tmplog to the mount command line, and inserts the text "tmplog" on a new command line.
-nolvm	Connects or creates a file system directly on a LUN without involving the host LVM.
-nopersist	Connects or creates a file system, or a snapshot that has a file system, without adding an entry in the host's file system table.

Option	Description
-reserve -noreserve	Used with snapdrive snap create storage or snapdrive snap connect commands to specify whether or not a SnapDrive creates a space reservation. By default, SnapDrive for UNIX creates a reservation for snap create storage operations, and does not create a reservation for snapdrive snap connect
-noprompt	Suppresses prompting during command execution. By default, any operation that might have dangerous or non-intuitive side effects prompts you to confirm that the SnapDrive for UNIX should be attempted. This option overrides that prompt; when combined with the -force option, SnapDrive for UNIX performs the operation without asking for confirmation.
-quiet (or -q)	Suppresses the reporting of errors and warnings, regardless of whether they are normal or diagnostic. It returns zero (success) or non-zero status. The -quiet option overrides the -verbose option.
-readonly	Required for configurations with Data ONTAP 6.5 or any configuration that uses traditional volumes. Connects the NFS file or directory with read only access Optional for configurations with Data ONTAP 7.0 that use FlexVol volumes. Connects the NFS file or directory tree with read only access. (Default is read/write).
-verbose (or -v)	Displays detailed output, where appropriate. All commands and operations accept this option, although some might ignore it.

Option	Description
`-unrelated`	Creates a snapshot of *file_spec* entities that have no dependent writes when the snapshot is taken. Because the entities have no dependent writes, SnapDrive creates a crash-consistent snapshot of the individual storage entities, but does not take steps to make the entities consistent with each other
`-vgsize (synonymous with -dgsize)`	Used with the `storage create` command to specify the size in bytes of the volume group you want to create.

Rules for keywords

SnapDrive for UNIX uses keywords to specify sequences of strings corresponding to the host and storage system objects with which you are working. The following rules apply to SnapDrive for UNIX keywords:

- Precede each keyword with a hyphen (-).
- Do not concatenate keywords.
- Enter the entire keyword and hyphen, not an abbreviation.

Command-line keywords

Here are the keywords you can use with the SnapDrive for UNIX commands. You use them to specify the targets of the SnapDrive for UNIX operations. These keywords can take one or more arguments.

Note

Some LVMs refer to *disk groups* and some refer to *volume groups*. In SnapDrive for UNIX, these terms are treated as synonyms. Moreover, some LVMs refer to *logical volumes* and some refer to *volumes*. SnapDrive for UNIX treats the term *host volume* (which was created to avoid confusing *host logical volumes* with *storage system volumes*) and the term *logical volume* as synonymous.

Keyword	Argument used with this keyword
`-dg` (synonymous with `-vg`)	The name of the host disk group. You can enter the name of either a disk group or a volume group with this option.

Keyword	Argument used with this keyword

`-destdg`
`-desthv`
`-destlv`
`-destvg`
	The destination group or volume.

`-file`	The name of a NFS file.
`-filer`	The name of a storage system.
`-filervol`	The name of the storage system and a volume on it.
`-fs`	The name of a file system on the host. The name used is the directory where the file system is currently mounted or is to be mounted (i.e., the mount point).
`-hostvol` (synonymous with `-lvol`)	The host volume name, including the disk group that contains it. For example, you might enter large_vg/accounting_lvol.

Keyword	Argument used with this keyword
`-igroup`	The name of an initiator group (igroup). It is strongly recommended that you use the default igroup that SnapDrive for UNIX creates instead of specifying an igroup on the target storage system. The default igroup is hostname_protocol_SbIg where: ◆ *hostname* is the local (non-domain qualified) name of the current host. ◆ *protocol* is either `fcp` or `iscsi`, depending which protocol the host is using. If the igroup hostname_protocol_SbIg does not exist, SnapDrive for UNIX creates it and places all the initiators for the host in it. If it exists and has the correct initiators, SnapDrive for UNIX uses the existing igroup. If the igroup exists, but does not contain the initiators for this host, SnapDrive for UNIX creates a new igroup with a different name and uses that igroup in the current operation. To avoid using the same name, SnapDrive for UNIX includes a unique number when it creates the new name. In this case, the name format is hostname-number_protocol_SbIg. If you supply your own igroup name, SnapDrive for UNIX does not validate the contents of the igroup. This is because it cannot always determine which igroups are on the hosts. For details on specifying igroups, see the SnapDrive for UNIX man page.

Keyword	Argument used with this keyword
`-lun`	The name of a LUN on a storage system. For the first LUN name you supply with this keyword, you must supply the full pathname (storage system name, volume name, and LUN name). For additional LUN names, you can specify either only the names within their volume (if the volume stays unchanged) or a path to indicate a new storage system name or a new volume name (if you just want to switch volumes).
`-lvol` (synonymous with `-hostvol`)	The logical volume name, including the volume group that contains it. For example, you might enter large_vg/accounting_lvol as the logical volume name.
`-snapname`	The name of a snapshot.
`-vg` (synonymous with `-dg`)	The name of the host volume group. You can enter the name of either a disk group or a volume group with this option.

Command-line arguments

The following table describes the arguments you can specify with the keywords.

Use the format `snapdrive type_name operation_name [<keyword/option> <arguments>]`; for example, if you wanted to create a snapshot called snap_hr from the host file system /mnt/dir, you would enter the following command line:

```
snapdrive snap create -fs /mnt/dir -snapname snap_hr
```

Argument:	Description:
dest_fspec	The name by which the target entity will be accessible after its disk group is connected.
dgname	The name of a disk group or volume group.
filername	The name of a storage system.

Argument:	Description:
filer_path	A pathname to a storage system object. This name can contain the storage system name and volume, but it doesn't have to if SnapDrive for UNIX can use default values for the missing components based on values supplied in the previous arguments. The following are examples of pathnames: test_filer:/vol/vol3/qtree_2 /vol/vol3/qtree_2 qtree_2

file_spec	The name of a storage entity such as a host volume, LUN, disk or volume group, file system or NFS directory tree.
	In general, you use the *file_spec* argument as one of the following:
	◆ An object you want SnapDrive for UNIX to take a snapshot of or to restore from a snapshot.
	◆ An object that you want to either create or use when provisioning storage.
	The objects do not have to be all the same type. If you supply multiple host volumes, they must all belong to the same volume manager.
	If you supply values for this argument that resolve to redundant disk groups or host volumes, the command fails.
	Example of incorrect usage: This example assumes dg1 has host volumes hv1 and hv2, with file systems fs1 and fs2. As a result, the following arguments would fail because they involve redundant disk groups or host volumes.
	`-dg dg1 -hostvol dg1/hv1`
	`-dg dg1 -fs /fs1`
	`-hostvol dg1/hv1 -fs /fs1`
	Example of correct usage: This example shows the correct usage for this argument.
	`-hostvol dg1/hv1 dg1/hv2`
	`-fs /fs1 /fs2`
	`-hostvol dg1/hv1 -fs /fs2`

Argument:	Description:
fspec_set	Used with the `snap connect` command to identify a ◆ Host LVM entity ◆ File system contained on a LUN The argument also lets you specify a set of destination names that SnapDrive for UNIX uses when it makes the entity available in the newly connected copy of the snapshot. The format of for *fspec_set* is: `{ -vg | -dg | -fs | -lvol | -hostvol } src_fspec [dest_fspec] [{ -destdg | -destvg } dg_name] [{ -destlv | -desthv } lv_name]` For more information, see the discussion for "Checklist for using the snapshot connect command" on page 259.

host_lvm_fspec	Lets you specify whether you want to create a file system, logical volume, or disk group when you are executing the `storage create` command. This argument may have any of the three formats shown below. The format you use depends on the entity you want to create.
	Note
	The `-dg` and `-vg` options are synonyms that reflect the fact that some operating systems refer to disk groups and others refer to volume groups. In addition, `-lvol` and `-hostvol` are also synonyms. This guide uses `-dg` to refer to both disk groups and volume groups and `-hostvol` to refer to both logical volumes and host volumes.
	To create a file system, use this format:
	`-fs file_spec [-fstype type] [-fsopts options] [-hostvol file_spec] [-dg dg_name]`
	To create a logical or host volume, use this format:
	`[-hostvol file_spec] [-dg dg_name] | -hostvol`
	To create a disk or volume group, use this format:
	`file_spec [-dg dg_name] | -dg dg_name`

Argument:	Description:
	You must name the top level entity that you are creating. you may not need to supply names for any underlying entities. If you don't supply names for the underlying entities, SnapDrive creates them with internally generated names. If you specify that SnapDrive for UNIX create a file system, you must specify a type that SnapDrive for UNIX supports with the host LVM. These types include the following: ◆ AIX: JFS2 ◆ HP-UX: VxFS ◆ Linux: Ext3 Current values that you can include with the `-fstype` argument are vxfs (HP-UX), jfs2 (AIX), or ext3 (Linux). The option `-fsopts` is used to specify options to be passed to whatever host operation creates the new file system (e.g., *mkfs*).
ig_name	The name of an initiator group.

long_filer_path	A pathname that includes the storage system name, volume name, and possibly other directory and file elements within that volume. The following are examples of long pathnames:
	
	`test_filer:/vol/vol3/qtree_2`
	
	`10.10.10.1:/vol/vol4/lun_21`

long_lun_name	A name that includes the storage system name, volume, and LUN name. The following is an example of a long LUN name: `test_filer:/vol/vol1/lunA`

Argument:	Description:

long_snap_name	A name that includes the storage system name, volume, and snapshot name. The following is an example of a long snapshot name:
	`test_filer:/vol/account_vol:snap_20040202`
	With the `snap show` and `snap delete` operations, you can use the asterisk (*) character as a wildcard to match any part of a snapshot name. If you use a wildcard character, you must place it at the end of the snapshot name. SnapDrive for UNIX displays an error message if you use a wildcard at any other point in a name.
	Example: This example uses wildcards with both the `snap show` command and the `snap delete` command:
	`snap show myfiler:/vol/vol2:mysnap*`
	`myfiler:/vol/vol2:/yoursnap*`
	`snap show myfiler:/vol/vol1/qtree1:qtree_snap*`
	`snap delete 10.10.10.10:/vol/vol2:mysnap*`
	`10.10.10.11:/vol/vol3:yoursnap* hersnap`
	Limitation for wildcards: You cannot enter a wildcard in the middle of a snapshot name. For example, the following command line produces an error message because the wildcard is in the middle of the snapshot name:
	`banana:/vol/vol1:my*snap`

lun_name	The name of a LUN. This name does not include the storage system and volume where the LUN is located. The following is an example of a LUN name: `lunA`
path	Any path name.

Argument:	Description:
short_snap_name	A name that consists of only the snapshot name. You can use the short form of the snapshot name when you do not need to include the storage system and storage system volume as part of the name. For example, when you create a snapshot, the storage system and volume must be the same as those of the original data. You can use wildcards with the `snap show` and `snap delete` operations. If you use a wildcard, you must place it at the end of the snapshot name. SnapDrive for UNIX displays an error message if you enter a wildcard in the middle of a snapshot name. **Example:** This example uses wildcards with the `snap show` command: `snap show toaster:/vol/vol1:mysnap*` **Limitations for wildcards:** You cannot enter a wildcard in the middle of a snapshot name. For example, the following snapshot name is incorrect: `toaster:/vol/vol1:test*snap`
size	The size in bytes of an object such as a LUN or a disk group. You can abbreviate sizes; for example, you might enter use 2g or 500m.
snap_name	A snapshot name in either the long or short form.
src_fspec	Identifies an LVM entity within the snapshots you want to connect to a different location using `snap connect`. This name is used to locate the disk group within the snapshot that needs to be connected in order to make the entity available.
volname	The name of the storage system volume.

Glossary

Attach Kit (HAK)

The attach kit or host attach kit (HAK) is the standard software and documentation to allow you to connect your host to a storage system.

Cluster

You can have a storage system cluster or a host cluster.

The standard storage system cluster involves having a pair of storage systems attached via fabric to a switch in such a way that one can serve its partner's data if the partner fails.

A host cluster refers to a host with multiple nodes. See the attach or support kit documentation for your operating system for information on the number of supported nodes on a host.

Clustered failover (CFO)

This is a method that ensures data availability by transferring the data service of a failed storage system to another storage system in the cluster. Transfer of data service is often transparent to users and applications

Disk group

A unit of storage that can contain one or more volumes and one or more file systems.

A set of LUNs managed by a Logical Volume Manager (LVM) on the host. LUNs in a disk group are combined in stripes, mirrors, and other RAID combinations by the host LVM. These combinations can then be divided into host volumes, which can be used to store data directly (as if they were raw disks), or can be used to host a file system. The data on a disk group is generally spread over all LUNs in the disk group, making it difficult to impossible to restore only a single LUN’s data. SnapDrive for UNIX uses term *disk group* interchangeably with the term *volume group*.

Failover

Failover refers to situations where a system component fails and another component takes over its functions while the system continues to operate. Failover can occur between storage system heads configured in a CFO cluster, among redundant FCP or iSCSI paths from the host to storage system LUNs, or from one clustered host to another.

In a clustered failover, an alternate system takes over and emulates the primary system if the primary system becomes unusable.

FCP initiator See Initiator.

file_spec Any supported object, like a host volume, disk group, file system or NFS file or directory tree, that snapshot uses to create a snapshot.

File system A file system in the SnapDrive for UNIX context refers to the file organization system supported by SnapDrive for UNIX on the host operating system.

Host A host is the machine running a UNIX operating system that accesses storage on a storage system.

Host bus adapter (HBA) A Host Bus Adapter (HBA) refers to the adapters used to connect hosts and storage systems in a SAN so that hosts can access LUNs on the storage systems using FCP. *See also* LUN (Logical Unit Number).

Host volume A host volume exists within a logical volume manager. It is a subset of a disk group (volume group) that acts as a disk to applications accessing it. The data in a host volume can be distributed over all or a subset of the disks in the disk group to which it belongs. Use of a host volume allows the administrator to easily resize the amount of storage assigned to a file system or application, without having to change the *disk* the application is using. The term *host volume* is used interchangeably with *logical volume*. The term *volume*, which is often used in conjunction with host logical volume managers, is avoided in this document, to avoid confusion with storage system volumes.

initiator The hardware that initiates data exchange.

initiator group (igroup)

Initiators that are grouped as authorized to access particular targets.

There are three categories of igroups that apply to SnapDrive for UNIX

- Foreign. This igroup contains only initiators not found on the local host.
- Local. This igroup contains only initiators found on the local host.
- Mixed. This igroup contains both initiators found on the local host and initiators not found there.

Logical volume

See host volume.

LUN (Logical Unit Number)

LUN refers to a logical unit of storage identified by a number.

LUN ID

This is the numerical identifier for a LUN.

LVM storage entities

Disk groups with or without host volumes and file systems that are created using the logical volume manager on the host.

Network Interface Card (NIC)

A network interface card (NIC) refers to a Gigabit Ethernet (commonly known as GbE) or a Fast Ethernet card that is compliant with the IEEE 802.3 standards.

These cards can provide the following connectivity functions:

- Connect hosts and storage systems to a local area network (LAN)
- Connect hosts and storage systems to data-center switching fabrics, specifically, enable hosts to connect to LUNs on storage systems using iSCSI

Raw Data

Data that is passed to an I/O device without being interpreted or processed. Raw access to a disk device implies that it is being used directly by an application to store data, rather than having a file system on it and letting the application store data in files in the file system.

Raw storage entities

LUNs or LUNs that contain file systems that are mapped directly to the host. Raw entities are created without using a Logical Volume Manager.

Rollback

This is a snapshot of the data on the storage system that SnapDrive for UNIX makes before it begins a `snap restore` operation. Having a rollback snapshot means that, in the event of a problem with a `snap restore` operation, you can restore the data on the storage system to the state it was in before the operation began. This option is enabled in the snapdrive.conf file by default (see "Setting values in snapdrive.conf" on page 85).

SAN (Storage Area Network)

A SAN (Storage Area Network) is a storage setup composed of one or more storage systems and connected to one or more hosts in an FCP or an iSCSI environment.

To a host running SnapDrive for UNIX, a connected SAN is just another target storage device within which SnapDrive for UNIX can create and manage LUNs.

Snapshot

A snapshot refers to the technology that enables the recovery after accidental deletion or modification of the data stored on a storage system by referencing a point-in-time image of that data.

Storage entity

The storage object, like a host volume, disk group, file system or NFS file or directory tree, that Snapshot uses to create a snapshot.

storage system

A storage system (sometimes called a filer) is a storage system that supports the FCP (Fibre Channel Protocol), iSCSI, and/or GbE (Gigabit Ethernet) protocols.

storage system volume

A storage system volume is a functional unit of storage. It comprises a collection of physical disks. A volume can be composed of one or more RAID groups to ensure data integrity and availability if multiple disks fail simultaneously within the same volume. For more information about storage system volumes, see the Data ONTAP *Storage Management Guide*.

Support Kit The support kit is the standard software and documentation supplied to allow you to connect your host to a storage system using an iSCSI protocol.

Target port A target is the HBA port on the storage system used to receive the SCSI I/O commands that the host initiator port sends. *See also* Initiator.

Volume See either host volume or storage system volume.

Index

Symbols

<host>_info utility 220

A

access permissions 4
    control levels 105
    examples 109
    permissions file 105
    setting 105
    setting configuration option 68
    steps for setting 106
    steps for viewing 109
-addlun command option 292
AIX host
    example of successful installation 31
    example of successful uninstallation 38
    example of uninstall warning 39
    files installed 57
    installation steps for SnapDrive for UNIX 25
    requirements 24
    uncompressing downloaded file 25
    uninstalling SnapDrive for UNIX 33
    using SMIT 26
    verify installation 32
arguments
    file_spec 300
    filer_path 300
    long_filer_path 302
    path 303
    rules for using 299
    short_snap_name 304
    snap_name 304
attach kit
    definition 305
    install 21
audit log file 68, 89
    disabling 90
    example 93
    maximum size 69
    number of old files 69
    specifying pathname 90
    using 92
audit-log-file option 68
audit-log-save option 69
-autoexpand command option 292
-autorename command option 292
autosupport
    configuration options 70
    example of a message 98
    setting up 98
    snapdrive.conf option 70
    using 97
autosupport-enabled option 70
autosupport-filer 70
available-lun-reserve option 71

B

bytes
    adding to storage 195

C

CD-ROM
    getting software 23
CFO
    definition 305
checklist
    general information 246
    host disconnect command 281
    snap connect command 259
    snap create command 247
    snap delete command 257
    snap disconnect command 263
    snap rename command 250
    snap restore command 254
    snap show command 252
    storage connect command 275, 279
    storage create command 265
    storage delete command 284
    storage disconnect command 277
    storage resize command 269
    storage show command 271

cluster
definition of 305
host 305
storage system 305
support in SnapDrive for UNIX 7
clustered failover
definition 305
command line mode
using 100
commands
-addlun option 292
arguments 299
-autoexpand option 292
-autorename option 292
config access 65
config check luns 65
config delete 65
config list 64, 113
config prepare luns 65
config set 65
config show 64
-dg keyword 296
disconnect 162
file_spec arguments 300
filer_path argument 300
-force option 139, 292
-fsopts option 293
-fstype option 293
-full option 293
-growby option 293
-growto option 294
-help option 294
host connect 208
host disconnect 215
-hostvol keyword 297
-igroup keyword 298
keyword 296
keyword rules 296
long_filer_path argument 302
-lunsize option 294
-noprompt option 295
options 292
path argument 303
-quiet option 295
running from the command line 100
short_snap_name argument 304
snap connect 153
snap create 122, 123
snap delete 166
snap list 133
snap rename 134
snap restore 139
snap show 127
snap_name argument 304
snapdrive config delete 113
snapdrive config set 112, 113
snapdrive version 87
-snapname keyword 299
storage show 186, 187, 188, 192, 193
summary of all commands 288
swverify 44, 50
-verbose option 295
Compatibility and Configuration Guide for NetApp's FCP and iSCSI Products 13
config command
access 65
check luns 65
delete 65
list 64
options 64
prepare luns 65
set 65
show 64
config delete command
SnapDrive for UNIX 113
config list command 113
config list operation 64
config set command 112
SnapDrive for UNIX 113
config show operation 64
configuration options
all-access-if-rbac-unspecified 68
audit-log-file 68
audit-log-max-size option 69
audit-log-save 69
autosupport-enabled 70
autosupport-filer 70
available-lun-reserve 71
default-noprompt 71
default-transport 72

device-retries option 72
device-retry-sleep-secs 73
enable-implicit-host-preparation 74
filer-restore-retries 75
filer-restore-retry-sleep 75
filesystem-freeze-timeout 75
mgmt-retries 75
mgmt-retry-sleep 76
mgmt-sleep-long-secs 76
multipathing-type 76
password-file 77
path 77
prefix-filer-lun 77
prepare-lun-count 78
recovery-log-file 78
recovery-log-save 79
snapcreate-consistency-retries 80
snapdelete-delete-rollback-with-snap 80
snaprestore-delete-rollback-after-restore 81
snaprestore-make-rollback 81
snaprestore-must-make-rollback 82
snaprestore-must-make-snapinfo-on-qtree 79
snaprestore-snapmirror-check 82
space-reservations-enabled 83
trace-enabled 83
trace-level 83
trace-log-file 84
trace-log-save 85
use-https-to-filer 85
viewing with config show 64
configuring
storage system volumes 19
connecting snapshots
examples 157
restrictions 149
steps for connecting 153
conventions
formatting x
keyboard xi
creating snapshots
examples 125
snap create fails 223
steps for creating 122
using SnapDrive for UNIX 122
creating storage 172
Ctrl-C
distrupts snapshot restoration 138

D

data
changes after snap restore 225
raw 307
data collection utility
examples 222
executing 221
information collected 220
Data ONTAP
version 16
Data ONTAP Block Access Management Guide 14
default-noprompt option 71
default-transport option 72
deleting snapshots
examples 167
reasons 165
restrictions 165
steps 166
depot file
HP-UX installation 41
device-retries option 72
device-retry-sleep-secs option 73
-dg 296
disconnecting snapshots
examples 164
steps 162
disconnecting storage 202, 206
disk
data changes after restore 225
disk group
definition 305
increasing size 194
displaying snapshots
examples 129
steps for displaying 127
documentation
additional 13
Compatibility and Configuration Guide for NetApp's FCP and iSCSI Products 13
Data ONTAP Block Access Management

Guide 14
FCP host attach kit 14
iSCSI host support kit 14
SnapDrive for UNIX Compatibility Matrix 13
SnapDrive for UNIX man page 13
SnapDrive for UNIX Quick Start 13
SnapDrive for UNIX Release Notes 13
storage system 16
System Configuration Guide 14

E

enable-implicit-host-preparation option 74
error messages 227
device not correct LUN 229
limit on open files 229
return codes 231
examples 167
snap connect command 262
snap create command 248
snap delete command 258
snap disconnect command 264
snap rename command 251
snap restore command 256
storage connect command 276, 278, 280
storage create command 268
storage resize command 270
storage show command 273

F

failover
cluster 7
definition 305
FCP
license required 17
FCP host attach kit documentation 14
Fibre Channel Protocol *See* FCP
file system
creating 173
definition 306
file sytems
resizing 197
file_spec 300
filer
cluster 7
defined ix
filer *See* storage system
filer volume *See* storage system volume
filer_path
example 300
filer-restore-retries option 75
filer-restore-retry-sleep option 75
FilerView
setting snap reserve 20
files
.rmp (Linux) 48
<host>_info utlity 220
access permissions 105
audit log 68, 69, 89
audit log file 92, 93
audit-log-save 69
depot (HP-UX) 41
installed on hosts 57
limit open files 229
RBAC permissions 68
recovery log 79, 89
recovery log file 93, 95
sdhost-name.prbac 105
snapdrive.conf 64
snapdrive.dc 220
trace log 84, 85, 89
filesystem-freeze-timeout option 75
-force
command option 292
-fsopts
command option 293
-fstype
command option 293
-full command option 293

G

-growby
command option 293
growby
command option 195
-growto
command option 294
growto
command option 195

guidelines
creating storage system volumes 19

H

HAK (Host Attach Kit)
install 21
-help
command option 294
host
<host>_info utility 220
cluster 305
clustered 7
displaying storage information 188
limit on open files 229
non-originating 145
volume 306, 309
Host Attach Kit *See* HAK
Host Bus Adapter (HBA)
definition 306
host connect command
steps for using 208
host disconnect command
checklist 281
examples 215
steps for using 215
host entities
creating
storage create command
creating host entities 173
deleting 218
disconnecting LUNs 206
host volume
definition 306
hosts
AIX requirements 24
comparing platforms 10
differences in commands 288
differences in platforms 11
HP-UX requirements 40
installation steps on AIX host 25
installation steps on HP-UX host 41
installation steps on Linux host 48
installation steps on Solaris host 53
Linux requirements 47
preparing for LUNs 88
resizing 197
Solaris requirements 52
terminology differences 11
-hostvol 297
HP-UX host
depot file 41
example of successful installation 44
example of successful uninstall 46
files installed 57
installation steps for SnapDrive for UNIX 41
moving file to host 41
requirements 40
uninstalling SnapDrive for UNIX 45
HTTPS 3
enabling 115

I

-igroup keyword 298
increasing storage size 194
installation
verify 44
installing
completing the installation 56
steps for AIX host 25
steps for HP-UX host 41
steps for Linux host 48
steps for Solaris host 53
upgrading SnapDrive for UNIX 61
verify 44, 50
IP address
partner 17
iSCSI
license required 17
problem if halted 229
iSCSI host suppor kit documentation 14

K

keywords 296
-dg 296
-hostvol 297
-igroup 298
rules for using 296
-snapname 299

L

license
- FCP 17
- iSCSI 17
- storage system 17

Linux host
- .rpm file 48
- adding host entries for LUNs 88
- error if iSCSI halted 229
- example of successful installation 49
- files installed 57
- installation steps for SnapDrive for UNIX 48
- moving downloaded file 48
- preparing for LUNs 88
- requirements 47
- uninstalling SnapDrive for UNIX 51
- verify installation 50

log files
- audit 69
- audit log 68, 89, 92, 93
- disabling 90
- enabling 89, 90
- recovery 79
- recovery log 89, 93, 95
- specifying pathnames 90
- trace 84, 85
- trace log 89

Logical Unit Numbers *See* LUN
logical volume
- creating
  - disk group
    - creating 173

Logical volume manager *See*LVM ix
logical volumes
- definition 307

logins
- storage system 111

long_filer_path
- example 302

LUN ID
- defined ix
- definition 307

LUNs
- adding host entries for LUNs 88
- connecting to host 208
- connecting to storage 198
- defined ix
- definition 307
- deleting 216
- disconnecting from host entity 206
- disconnecting from storage 202
- disconnecting to host 215
- displaying storage information 186
- not recognized 229
- preparing host 88
- preparing the host 65

-lunsize
- command option 294

LVM
- using to resize 197

M

man page
- SnapDrive for UNIX 13

mgmt-retries option 75
mgmt-retry-sleep option 76
mgmt-sleep-long-secs option 76
multipathing-type option 76

N

name/value pair
- snapdrive.conf file 86, 116

Network Interface Card (NIC), definition 307
non-originating host 145
-noprompt
- command option 295

NOW description page
- most up-to-date requirements 7

NOW site
- downloading software 22

O

open files
- limit 229

operating system
- limit on open files 229
- storage system 16

options
-addlun 292
-autoexpand 292
-autorename 292
-force 139, 292
-fsopts 293
-fstype 293
-full 293
-growby 293
-growto 294
-help 294
--lunsize 294
-noprompt 295
-quiet 295
using on command lines 292
-verbose 295

P

partner
specify IP address 17
password-file option 77
passwords
storage system 111
path 303
path configuration option 77
prefix-filer-lun option 77
prepare-lun-count option 78

Q

Quick Start Guide 13
-quiet
command option 295

R

raw
defintion 307
RBAC *See* access permissions
recommendations
snap reserve 19
storage system volumes 19
recovery log file 79, 89
disabling 90
enabling 90
example 95
specifying pathname 90
using 93
recovery-log-fileoption 78
recovery-log-save option 79
Red Hat Linux host
see Linux host 47
renaming snapshots
examples 135
steps for renaming 134
requirements
AIX host 24
HP-UX host 40
Linux host 47
most current 7
Solaris host 52
storage system 16
restoring snapshots
examples 142
-force option 139
interrupting with Ctrl-C 138
restoring from a different host 145
restrictions 136
snap restore fails 223
steps for restoring 138
time required 138
warning 224
restrictions
on connecting snapshots 149
on restoring snapshots 136
snapshops 122
return codes
error messags 231
rollback 80, 81, 82
definition 308
rpm file
Linux installation 48
rsh
not used 4

S

SAN (Storage Area Network), definition 308
security
considerations 3

features 104
short_snap_name 304
SMIT
using on AIX 26
using on AIX host 25
snap connect command 153
checklist 259
examples 157, 262
snap create command 122, 123
checklist 247
examples 125, 248
fails 223
snap delete command 166, 167
checklist 257
commands
snap delete 165
example 258
snap disconnect command 162
checklist 263
examples 164, 264
snap rename command 134
checklist 250
examples 135, 251
snap reserve 19
recommended value 18
resetting 19
snap restore command 139
checklist 254
data on disk changes 225
examples 142, 256
fails 223
warning 224
snap show command
checklist 252
examples 129
snap_name 304
snapcreate-consistency-retries option 80
snapdelete-delete-rollback-with-snap option 80
SnapDrive for UNIX
access permissions 4, 105
additional documentation 13
AIX requirements 24
audit log 92
autosupport 97
capabilities 2
cautions 18
checking version 87
command arguments 299
command keywords 296
command options 292
comparing platforms 10
config access 65
config check luns 65
config delete 65
config delete command 113
config list 64
config prepare luns 65
config set 65
config set command 112, 113
config show 64
configurations 7
connecting snapshots 147
creating snapshots 122
data collection utlity 220
deleting snapshots 165, 166
deleting user logins 113
disabling log files 90
disconnecting snapshots 160
displaying information on storage 185
displaying snapshots 127
downloading software 22
enabling log files 89
error messages 227
files installed on host 57
getting software 21
getting software from CD-ROM 23
how it works 2
HP-UX requirements 40
HTTPS 3
installation steps on AIX 25
installation steps on HP-UX 41
installation steps on Linux 48
installation steps on Solaris 53
Linux requirements 47
log files 89
multiple platforms 10
prerequisites 5
recommendations 18
recovery log 93
renaming snapshots 134

requires attach kit 21
restoring snapshots 136
running from the command line 100
security 3
security features 104
setting snap reserve 19
setting up autosupport 98
setting values in snapdrive.conf 85
snapshot
created with SnapDrive for UNIX 2
snapshot restrictions 122
Solaris requirements 52
specifying log file pathnames 90
specifying logins for storage system 111
stack 6
storage 2
storage system volumes 19
summary of all commands 288
supported snapshots 118
time to restore snapshots 138
troubleshooting tips 223
uninstalling from AIX host 33
uninstalling from HP-UX host 45
uninstalling from Linux host 51
uninstalling from Solaris host 54
upgrading versions 61
version command 87
volume manager terminology 11
SnapDrive for UNIX Compatibility Matrix 13
SnapDrive for UNIX Quick Start 13
SnapDrive for UNIX Release Notes 13
snapdrive.conf
all-access-if-rbac-unspecified option 68
audit-log-file option 68
audit-log-max-size option 69
audit-log-save option 69
autosupport-enabled option 70
autosupport-filer option 70
available-lun-reserve option 71
changes with version 2.0 61
default values 65
default-noprompt option 71
default-transport option 72
device-retries option 72
device-retry-sleep-secs option 73
enable-implicit-host-preparation option 74
filer-restore-retries option 75
filer-restore-retry-sleep option 75
filesystem-freeze-timeout option 75
mgmt-retries option 75
mgmt-retry-sleep option 76
multipathing-type option 76
name/value pair 86, 116
password-file option 77
path option 77
prefix-filer-lun option 77
prepare-lun-count option 78
recovery-log-file option 78
recovery-log-save option 79
setting values 85
snapcreate-consistency-retries option 80
snapdelete-delete-rollback-with-snap option 80
snaprestore-delete-rollback-after-restore option 81
snaprestore-make-rollback option 81
snaprestore-must-make-rollback option 82
snaprestore-must-make-snapinfo-on-qtree option 79
snaprestore-snapmirror-check option 82
space-reservations-enabled option 83
trace-enabled option 83
trace-level option 83
trace-log-file option 84
trace-logs-ave option 85
use-https-to-fileroption 85
viewing 64
snapdrive.dc 220
examples 222
executing 221
tasks performed 220
SnapDriver for UNIX
snapshots 2
-snapname 299
SnapRestore
license required 17
snaprestore-delete-rollback-after-restore option 81
snaprestore-make-rollback option 81
snaprestore-must-make-rollback 82
snaprestore-must-make-snapinfo-on-qtree option

79
snaprestore-snapmirror-check option 82
snapshot
about 118
creating 122
definition 308
deleting 165, 166
disconnecting 160
displaying information 127
examples of snap connect command 157
examples of snap create command 125
examples of snap delete command 167
examples of snap disconnect command 164
examples of snap rename command 135
examples of snap restore command 142
examples of snap show command 129
interruping restoration 138
listing on storage system 133
listing with wildcards 133
reasons to delete 165
renaming a snapshot 134
restoring 136
restoring from a different host 145
restrictions 122
restrictions on connecting 149
restrictions on deleting 165
restrictions on restoring 136
rollback 80, 81, 308
rollback option 82
spanning storage systems 119
spanning volumes 119
steps for connecting 153
steps for displaying 127
steps for renaming 134
steps for restoring 138
time to restore 138
using wildcards 304
snapshots
connecting from different locations 147
software
downloading 22
getting 21
getting from CD-ROM 23
uncompressing 25
uncompressing on Solaris 53
Solaris host
adding host entries for LUNs 88
files installed 57
installation steps for SnapDrive for UNIX 53
preparing for LUNs 88
requirements 52
uncompressing SnapDrive for UNIX software 53
uninstalling SnapDrive for UNIX 54
space reservations 18
space-reservations-enabled option 83
special messages xi
stack
required 6
storage
connecting 198
connecting LUNs to host 208
created with SnapDrive for UNIX 2
creating 172
deleting 216, 217, 218
disconnecting 202, 206
disconnecting LUNs to host 215
disconnecting specific LUNs 204
displaying information 185
examples of storage show command 189
growing 195
resizing 194
steps for deleting host entities 218
steps for deleting LUNs 217
steps for displaying information 186
steps for displaying information on host entities 188
steps for displaying information on storage systems 187
Storage Area Network *See* SAN
storage commands
using across storage systems 171
storage connect command
checklist 275, 279
examples 199, 200, 201, 209, 210, 276, 278, 280
steps for using 199
storage create command
checklist 265
examples 184, 268

181
storage delete command 217, 218
checklist 284
limitations 216
storage disconnect command 206
checklist 277
examples 204, 205, 206, 213, 214
limitations 202, 211
steps for using 204
tips 203
storage provisioning 170
storage resize command 195
checklist 269
examples 195, 270
limitations 194
steps for using 195
storage show command 186, 187, 188, 192, 193
checklist 271
examples 189, 273
restrictiosns 186
storage system
cluster 305
clustered 7
configuring volumes 19
definition 308
deleting SnapDrive for UNIX logins 113
displaying storage information 187
documentation 16
failover 7
guides for creating volumes 19
login required 3
logins 111
operating system 16
preparing 16
required license 17
requirements 16
resetting snap reserve 19
setup program 17
specify login information 111
specify partner IP address 17
target 309
volume 309
volume preparation 19
storage system volume
definition 308
storage systems
spanning 119, 171
verifying user names 113
support kit
definition 309
swverify command
verify installation 44, 50
System Configuration Guide 14

T

target
definition 309
Terminology ix
trace log file 84, 85, 89
disabling 90
enabling 90
specifing pathname 90
trace-enabled option 83
trace-level option 83
trace-log-file option 84
trace-log-save option 85
troubleshooting
data collection utility 220
data on disk changes 225
device not recognized 229
error message resturn codes 231
error messages 227
iSCSI halted 229
limits on open files 229
snap create fails 223
snap restore fails 223
tips 223

U

uninstalling
SnapDrive for UNIX from AIX host 33
SnapDrive for UNIX from HP-UX host 45
SnapDrive for UNIX from Linux host 51
SnapDrive for UNIX from Solaris host 54
upgrading
SnapDrive for UNIX 61
use-https-to-filer option 85
user logins
deleting from SnapDrive for UNIX 113

displaying names with config list 113
utilities
<host>_info 220
data collection 220

V

-verbose
command option 295
version command
SnapDrive for UNIX 87
volume
logical 307
volume group
increasing size 194
volumes
definition 309
host 306
preparing for storage system 19
recommendations for storage system 19
spanning 119
storage system 19, 308

W

wildcards
example 303
using with snapshots 304